NOTE TO USERS

The original manuscript received by UMI contains pages with broken and indistinct print. Pages were microfilmed as received.

This reproduction is the best copy available

UMI

University of Alberta

Analysis of Elastic and Elasto-plastic Behaviour of Spatial Structures

by

Amir H. Ghasemi Ghamsari ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Department of Mechanical Engineering

Edmonton, Alberta
Fall 1998

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-34769-9

Canada

University of Alberta

Library Release Form

Name of Author: Amir H. Ghasemi Ghamsari

Title of Thesis: Analysis of Elastic and Elasto-plastic Behaviour of Spatial Structures

Degree: Doctor of Philosophy

Year this Degree Granted: 1998

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly, or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Amir H. Ghasemi G.

Amir H. Ghasemi Ghamsari
#B4, 8504 98 St.,
Edmonton, Alberta, Canada
T6E 3M3

Date: Sept. 8, 1998

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Analysis of Elastic and Elasto-plastic Behaviour of Spatial Structures** submitted by **Amir H. Ghasemi G.** in partial fulfillment of the requirements for the degree of Doctor of Philosophy

David Steigmann

Dr. D. J. Steigmann (Supervisor)

Dr. S. A. Lukasiewicz

Dr. J. K. Szymanski

Gary Faulkner

Dr. M. G. Faulkner

Dr. A. Mioduchowski

Ruchong Qing

Dr. C. Q. Ru

Date: Sept. 2. 1998

Abstract

The nonlinear deformation of lattices with linear and nonlinear materials is analyzed employing the Dynamic Relaxation (DR) method to solve the equations of equilibrium. A new approach of considering the exact equations of equilibrium is presented and the equilibrium configurations are obtained as the steady state part of dynamical problems. Since all quantities may be treated as vectors, thus eliminating the need for the calculation of a stiffness matrix and its inverse, DR offers an attractive alternative. This solution technique is applied in both pre and post-buckling analyzes for a variety of different lattices. In the DR method the static solution represents the steady state part of the dynamical response and equilibrium solutions obtained by DR could be regarded as asymptotically stable. DR is applied to a variety of example problems in which linear elastic materials are considered in two ways:

(1) load control in which DR does not pick up the unstable branch on which descending load accompanies increasing displacement.

(2) displacement control which traces the whole load-displacement (stress-strain) curve as the stable solution.

The DR method is ideal for the case of large deformations (linear and non-linear materials), which include limit points or regions of very soft stiffness characteristics. The numerical procedure is also applied to include the analysis of nonlinear materials. The inelastic behavior of trusses is accounted for by tracing the complete stress-strain relationships in the elastoplastic range. The unloading characteristics of trusses in the post-critical elastic-plastic

stage are determined and the irreversibility of plastic strains are considered through an incremental displacement procedure.

Where previous analytical, numerical or experimental results are available, the present results for both linear and nonlinear materials are shown to compare favorably.

Dedication

To my late mother (God bless her), my father, and my sister and brother

Acknowledgement

I am especially grateful to Dr. D.J. Steigmann for his valuable guidance and support throughout the course of this work. I would like to take this opportunity to sincerely thank Dr. M.G. Faulkner and Dr. A. Mioduchowski for their help at various times. I would also like to thank Dr. A. Atai and Dr. D. Raboud for their help. I would also like to thank the partial financial support of Ministry of Culture and Higher Education of Iran and Natural Sciences and Engineering Research Council of Canada. Last but not the least, I would like to thank my late mother (God bless her) and my father who were always great support for me.

Contents

1 **Introduction** 1
1.1 Mathematical Formulation . . . 2
1.2 Solution Techniques for Tracing the Post-Buckling Path . . . 2
1.3 Inelastic Materials . . . 6
1.4 Thesis Outline . . . 6

2 **Formulation of the Problem** 8
2.1 Kinematics and the Basic Constitutive Hypothesis . . . 9
2.2 The Variation of the Potential Energy . . . 13
2.3 Lattices . . . 16
2.4 Equilibrium Equations for Small Deformation . . . 23
2.5 Numerical Results . . . 26

3 **Numerical Method (Dynamic Relaxation)** 40
3.1 Introduction . . . 40
3.2 Theory of Dynamic Relaxation . . . 41
3.3 Application of DR for Linear Deformations . . . 43
3.4 Numerical Results . . . 46

4 **Large Deformation of Structures by DR** 51

4.1 Dynamics of Rods for Large Deformation in Space 51
4.2 Two-Dimensional Large Deformation (Using DR Method) 57
4.3 Three-Dimensional Large Deformation (Using DR Method) 59
4.4 Numerical Results 62

5 Inelastic Post-Buckling Analysis of Truss Structures by DR126
5.1 Introduction 126
5.2 Failure of conventional DR for inelastic material and introducing incremental DR 128
5.3 Formulation of the Problem 132
5.3.1 Constitutive Laws 132
5.4 DR Algorithm for Inelastic Materials 138
5.5 Numerical Results 139

6 Concluding Remarks 170
6.1 Summary and Conclusions 170

A SOURCE CODE 179

List of Tables

4.1 Geometry at the end of a curved cantilever under the action of a dead tip load 118
4.2 The numerical result for comparison of NR and DR methods for nonlinear deformation of circular curved cantilever beam. . 122

List of Figures

2.1 Linear deformation for space truss 28
2.2 Linear out of plane deformation for plane frame 29
2.3 Linear deformation for space truss 31
2.4 Comparison of different boundary conditions for out of plane deformation 33
2.5 Comparison of different boundary conditions for in plane deformation 35
2.6 Linear out of plane deformation for plane frame 38
2.7 Linear out of plane deformation for plane frame 39

3.1 Deformed configuration of the plane frame using *FDM* with *DR* 47
3.2 Deformed configuration of the space truss using *FEM* with *DR* 49

4.1 Nonlinear deformation of a cantilever beam with DR. 64
4.2 Nonlinear deformation of two-member truss with DR. 67
4.3 Nonlinear deformation of two-member frame with DR. 71
4.4 Nonlinear deformation of tower truss with DR. 78
4.5 Analysis of snap through for truss-arch with DR. 80
4.6 Analysis of snap through for Fixed-fixed shallow circular arch with midspan concentrated vertical load. 82
4.7 The snap through of space structure with one central load with DR. 84

4.8 The snap through of space structure with symmetric loading with DR. 89
4.9 The snap through of space structure with non-symmetric loading with DR. 93
4.10 The snap through of planar structure with DR. 98
4.11 The snap through of planar structure with DR. 107
4.12 The snap through of space frame with DR. 115
4.13 The deformation of initially curved cantilever beam. 117
4.14 The deformation of initially circular curved cantilever beam. . 119
4.15 The buckling of a cantilever beam with NR. 120
4.16 The comparison of NR and DR methods for nonlinear deformation of circular curved cantilever beam. 121
4.17 The nonlinear deformation of fixed fixed semicircle under prescribed displacements in the middle.(using DR method) 123
4.18 The nonlinear deformation of a structure with two circular arcs (two 1/8 of a circle) under prescribed displacements in the middle.(using DR method) 124
4.19 The nonlinear deformation of fixed free semicircle under prescribed displacements. (using DR method) 125

5.1 Figures used to justify using incremental-DR instead of conventional DR. 130
5.2 Stress-strain relation for different materials 134
5.3 Coding for different paths of elastic-hardening and nonlinear-elastic 135
5.4 Flow chart for tracing the behaviour of elastic-hardening and nonlinear-elastic materials 136
5.5 Comparison of behaviour of different materials for two-member truss. 141
5.6 Comparison of behaviour of different materials for space truss. 152

Chapter 1

Introduction

There are many structures of practical engineering importance which can be modeled as long slender rods that undergo large three-dimensional deformations due to combined bending and twisting. A large deformation is one in which the initial geometry is changed significantly during the loading. Much work has been concentrated on the nonlinearities produced from changes of the geometry, because these types of structures are more likely to fail due to instability before significant nonlinear material response is observed. The elastic stability analysis of structures is concerned with the calculation of certain critical states at which the stiffness of the structure, with respect to a small disturbing force, becomes singular.

Apart from the geometric nonlinearity due to the inherently large deformations these structures can undergo, nonlinearity may be introduced into the problem through the materials used. Such structures could also have a material nonlinearity which adds complexity to the problem. As a result we need a method that is stable and works well for all situations.

Dynamic Relaxation (DR) is an explicit iterative method developed for static analysis of structural mechanics problems. This thesis is concerned with the application of DR to obtain solutions for structures undergoing large (nonlinear) three-dimensional deformations considering different materials including some exhibiting inelastic behavior.

1.1 Mathematical Formulation

The equations of equilibrium governing the three-dimensional deformations of long slender rods are well known. In his treatise on elasticity, Love [19] presents a derivation of these equations based on that of Kirchhoff (1859). In this presentation, the general case of a rod with two distinct principle axes in the cross section is assumed. Extensibility is taken into account as well. Landau and Lifshitz [16] derived the equations of equilibrium of the rod based on an analysis of an arbitrary infinitesimal section of the rod. These equations were expressed in terms of the tangent vector and its derivative. Steigmann and Faulkner [36] obtained the equilibrium equations for a rod using variational calculus to minimize the potential energy of the deformed rod. The rod was considered to be a one dimensional continuum with a strain energy which depends on the bending and twisting along the rod. The equations obtained were identical to those of Landau and Lifshitz [16]. Steigmann [37] introduced a theory for spatial lattices in a variational setting and derived necessary conditions restricting stable deformations. He obtained the equilibrium equations for a lattice with extensible rods using variational calculus to minimize the potential energy of the deformed lattice. This work will be the starting point for what follows in this thesis. The modeling and the mathematical formulation constitutes an important stage in the analysis of a structure, but of equal importance is the use of efficient procedures for the solution of the resulting system of equations.

1.2 Solution Techniques for Tracing the Post-Buckling Path

General computational techniques for the analysis of large deformations of spatial structures have been extensively studied in recent years. The most widely used method for the study of geometrically nonlinear problems is the Newton-Raphson method. In the Newton type methods, a lot of effort has been devoted to the correct formulation of the linearized incremental stiffness matrix from the governing nonlinear equations. The main drawback of the procedure is the considerable computational effort to calculate the tangent stiffness matrix by numerical or analytical differentiation at each iterative step. There is also a difficulty in surpassing critical points where the tangent stiffness matrix becomes singular. Although the methods based on the minimization of the total potential energy overcome these difficulties, running times and computer storage requirements become prohibitive for large problems. In recent years several strategies have been proposed to trace the nonlinear response from the pre-limit into the post-limit range. Among these are simple methods for suppressing equilibrium iterations as follows:

- *Method of artificial springs.*
  In this method, *artificial spring stiffness* are added to the stiffness matrix, to make it positive-definite through the load-deflection range [35]. A *load reduction factor* is then applied to get the actual external load on the structure. For simple structures and load patterns, this method is very attractive as it requires the addition of only one spring. However, for multi-degree-of-freedom problems, this method does not seem to be mathematically justifiable. Also, the addition of artificial springs destroys the banded nature of the stiffness matrix.

- *Displacement incrementation method*
  The most often used method to avoid the singularity at the critical points is the interchange of dependent and independent variables. Here a single displacement component selected as a controlling parameter is prescribed and the corresponding load level is taken as unknown. The method was initially proposed by Argyris [2] and modifications were suggested by Pian and Tong [26]. A valuable contribution in this field has been made by Batoz and Dhatt [5] as they have proposed an incremental displacement algorithm which does not destroy the banded

nature of the stiffness matrix. Powell [27] generalized this procedure of displacement incrementation. From the pre-buckling analysis of the structure, a monotonically increasing controlling displacement component is selected and will be incremented. Although in most practical structures it may not be difficult to identify this *controlling displacement*, for large structures the choice may not be obvious.

- *Arc-length methods.*

  It has been found that instead of incrementing a single displacement component as in the displacement incrementation method, it is advantageous to increment the total displacement vector itself. To this end, a parameter known as *arc-length* has been defined which limits the load and displacement steps. A *constrained equation* for arc length is written in addition to the N equilibrium equations to solve for N displacements and one load parameter.

  Riks [32] and Wempner [40] have proposed a constraint equation which fixes the arc length in $(N+1)$-dimensional space. The iterations are carried out on a plane normal to the tangent of the load-deflection curve. However, the use of this constraint equation, in addition to the equilibrium equations in the form suggested destroys the banded nature of the stiffness matrix. This difficulty could be circumvented by the two-step technique of Batoz and Dhatt [5]. Crisfield [8] has proposed a constraint equation that fixes the arc length in N-dimensional displacement space. The iterations are carried out on a *sphere*, over which the arc length is constrained to lie. The banded nature of stiffness matrix is retained using Batoz and Dhatt's algorithm.

Another approach, the segmental shooting technique, was developed for applications including the laying of offshore pipelines (Faulkner and Stredulinsky [11]) and the prediction of force systems produced by orthodontic reaction appliances (Lipsett *et al.*[17]; Faulkner *et al.*[12]). This technique avoids the direct solution of the nonlinear boundary value problem by considering the rod as being comprised of a large number of segments, each of which experiences only small displacements so that a linear solution can be applied over each segment. The total nonlinear solution is obtained by assembling the segments together. This formulation was found to be well suited to investigate the development of multiple solutions (Faulkner *et al.*[13]; Lipsett et al.[18]). This approach has been modified to take fully three-dimensional deformations into account (Raboud *et al.*[28];Raboud *et al.*[29] and Raboud *et al.*[30]).

An explicit vector method widely and successfully applied to a nonlinear analysis of structural mechanics in general is Dynamic Relaxation (DR). The object of this method is to trace step-by-step, for small time increments Δt, the motion of a structure from the time when it is initially loaded to the time when, due to imposed damping, it reaches a steady equilibrium state. In this method, the solution is found as the steady state part of the response of the dynamic problem formed by considering the equations of motion of the structure. The DR method is a stable method and does not require the computation or formulation of any tangent stiffness matrix.

In this field there is no proof of dynamical stability of the equilibrium configurations and also no proof of Lyapunov stability. However we can claim that the equilibrium configurations that we calculate are necessarily asymptotically stable with respect to dynamic chosen.

This method will be used for solving the nonlinear equations of equilibrium (with both linear and nonlinear materials). A standard central difference scheme is applied to the resulting nonlinear equations of motion and are recursively solved to obtain the equilibrated displacement field under a given loading or prescribed displacement.

1.3 Inelastic Materials

A realistic treatment of the behaviour of structures should include material nonlinearities together with geometric nonlinearities. In solving the equilibrium equations for inelastic materials the main principle to be considered is the path-dependent nature of the stress-strain curve during loading and unloading. Incremental-DR is a method which could be used for structures subjected to both material and geometrical nonlinearities with path dependent behaviour. In this technique, loading and prescribed displacements are considered in a incremental way. At the beginning and end of each increment, the structure is equilibrated by solving the nonlinear equations of equilibrium using DR.

In this work we apply DR to investigate the geometric and material nonlinearities of space trusses. When dealing with path-dependent problems, the incremental-DR technique is used, noting that in each iteration of DR, the stress state should be compared with that of the beginning of the increment to determine whether the structure is being loaded or unloaded. We explain more about incremental-DR in chapter 5. The behaviour of two structures considering Nonlinear elastic, Elastic-perfectly plastic and Elastic-hardening materials is studied and irreversibility of the strains under plastic loading is investigated.

1.4 Thesis Outline

Our contribution is that we applied variational procedure to derive the exact equation of equilibrium for the spatial lattices and used finite difference scheme to discretize them. Then we used Dynamic Relaxation method to solve for the deformed configuration. This technique has been used for nonlinear static analysis of other structures, such as cable networks and membranes. We used this method to analyze the behavior of a variety of structures consisting of extensible rods and derived some interesting results. We also modified Dynamic Relaxation and used the incremental Dynamic Relaxation method to study the large deformation of inelastic response of structures and elasto-plastic response of nonlinear materials with path-dependent behavior. Now we give a brief description of each of the following chapters of the thesis.

Chapter 2 describes some of the background theory concerning the kinematical and constitutive foundations for three-dimensional deformations of extensible spatial rods. Also, the equilibrium conditions at the nodes of the lattice associated with various types of nodal constraints are derived. The equilibrium equation for linear deformations of elastic materials are derived. As well, some general results are presented for linear elastic deformations.

In Chapter 3, the Dynamic Relaxation method is presented. The application of DR to linear deformations employing the Finite Difference Method and the Finite Element Method is given. Some results are presented using DR to verify the numerical procedure.

Chapter 4 considers the dynamics of rods for large deformations in space. The DR is then formulated to solve the nonlinear equilibrium equations for nonlinear elastic materials in two and three-dimensional deformations. Numerical results are then presented for a variety of problems. These are compared with previous analytical, numerical or experimental results to assess the effectiveness of the numerical technique.

Chapter 5 considers an application of the DR to include materials with inelastic behavior . Failure of conventional DR for inelastic material and introducing incremental DR is explained here. Some numerical results are also presented to illustrate the application of the DR algorithm.

Chapter 2

Formulation of the Problem

In this chapter, we use the nonlinear variational theory for three-dimensional deformation of extensible spatial rods. This derivation closely follows the work done by Steigmann [37] since the subsequent work is based on his work. This theory is based on the notion of a one-dimensional continuum endowed with a kinematical and constitutive structure sufficient to represent the dominant features of the mechanics of suitably loaded thin three-dimensional elastic bodies. This theory belongs to a general class, ranging from flexible cables to rods with significant flexural and torsional rigidity, that has received considerable attention. An extensive account of such theories and their mathematical structure may be found in Antman's book [1].

In this chapter, section (2.1) is devoted to a brief discussion of the kinematical and constitutive foundations of the theory under consideration. The framework used here generalizes the viewpoint adopted in [36] for inextensible rods. In section (2.2) the expressions for the first and second variations of the strain energy are presented based on a local characterization of kinematically admissible configurations.

The results are used in section (2.3) to obtain the Euler equations of equilibrium. These are identical in form to the well known statical equations of rod theory. Also the equilibrium conditions at the nodes of the lattice associated with various types of nodal constraints are derived. These, too, correspond to the conditions that would be obtained from elementary considerations. In section (2.4) the equilibrium equations for linear deformation of an initially straight rod with isotropic material and circular cross-section are derived using this theory.

2.1 Kinematics and the Basic Constitutive Hypothesis

Configurations of spatial rods are defined by mappings of an arclength parameter $S \in [0, L]$ onto $\{\mathbf{r}(S), \mathbf{e}_i(S)\}; i = 1, 2, 3$, where

- S is the initial arclength in a reference placement that varies in the domain $[0, L]$
- $s(S)$ is the deformed arclength that varies in the domain $[0, l]$
- $\hat{\mathbf{r}}(s) = \hat{\mathbf{r}}(s(S)) = \mathbf{r}(S)$ is the position vector of points on the rod in the deformed configuration relative to a fixed origin
- $\mathbf{e}_i(S)$ are the vector-valued functions that specify the orientations of *cross-section* in the deformed configuration

In the reference configuration the functions $\mathbf{r}(S)$ and $\mathbf{e}_i(S)$ take the values $\mathbf{X}(S)$ and $\mathbf{E}_i(S)$, respectively.

We take $\mathbf{E}_i(S)$ to be an orthonormal set in which $\mathbf{E}_1(S)$ is the unit tangent to the space curve defined by $\mathbf{X}(S) : \mathbf{E}_1(S) = \mathbf{X}'(S)$. Henceforth prime means derivative with respect to S. The vectors $\mathbf{E}_2(S)$ and $\mathbf{E}_3(S)$ span the plane normal to the curve at arclength station S. In principle, there is no further restriction on the specification of these vectors, but it is frequently advantageous to define them so that $\mathbf{E}_1 \cdot \mathbf{E}_2 \times \mathbf{E}_3 = 1$ with $\mathbf{E}_2(S)$ and $\mathbf{E}_3(S)$ along the geometric principal axes of the *cross-section.* We consider a constrained theory in which the set $\{\mathbf{e}_i(S)\}$ is orthonormal at every cross-section, with $\mathbf{e}_1 \cdot \mathbf{e}_2 \times \mathbf{e}_3 = 1$, and also let $\mathbf{e}_1(S)$ coincide with the unit tangent to the space curve defined by $\mathbf{r}(S)$. Thus we get $\mathbf{e}_1 = \frac{d\mathbf{r}}{ds}$, comparing to $\mathbf{E}_1 = \frac{d\mathbf{X}}{dS}$. We then have

$$\mathbf{r}' = \frac{d\mathbf{r}}{ds} \cdot \frac{ds}{dS} \tag{2.1}$$

$$\text{or} \qquad \mathbf{r}' = \lambda \mathbf{e}_1 \tag{2.2}$$

where $\lambda = \frac{ds}{dS} = |\mathbf{r}'|$ is the local *stretch* of the rod. Note that in the unstretched configuration $\lambda(S) = 1$.

The deformation gradient is defined by

$$d\mathbf{r}(S) = \mathbf{F} d\mathbf{X}(S) \tag{2.3}$$

We can write

$$d\mathbf{r} = \mathbf{r}'(S) dS \tag{2.4}$$

or

$$d\mathbf{r} = \lambda \mathbf{e}_1 dS \tag{2.5}$$

But since $d\mathbf{X} = \mathbf{X}'(S)dS = \mathbf{E}_1 dS$, so

$$dS = \mathbf{E}_1 \cdot d\mathbf{X} \tag{2.6}$$

Substitution Eqn.(2.6) into Eqn.(2.4) gives

$$d\mathbf{r} = \lambda \mathbf{e}_1 \left(\mathbf{E}_1 \cdot d\mathbf{X}\right) \tag{2.7}$$

or

$$d\mathbf{r} = (\lambda \mathbf{e}_1 \otimes \mathbf{E}_1) d\mathbf{X} \tag{2.8}$$

Comparing Eqns.(2.3) and (2.8) gives the deformation gradient as

$$\mathbf{F} = \lambda \mathbf{e}_1 \otimes \mathbf{E}_1 \tag{2.9}$$

or

$$\mathbf{F} = \mathbf{r}'(S) \otimes \mathbf{E}_1 \tag{2.10}$$

Moreover, the constraints on $\{\mathbf{e}_i\}$ and $\{\mathbf{E}_i\}$ imply the existence of a *rotation* $\mathbf{R}(S)$ ($\det\mathbf{R} = 1$, $\mathbf{R}^T = \mathbf{R}^{-1}$) such that

$$\mathbf{e}_i(S) = \mathbf{R}(S)\mathbf{E}_i(S) \tag{2.11}$$

In particular,

$$\mathbf{R} = \mathbf{R}\mathbf{1} = \mathbf{R}(\mathbf{E}_i \otimes \mathbf{E}_i) = (\mathbf{R}\mathbf{E}_i) \otimes \mathbf{E}_i = \mathbf{e}_i \otimes \mathbf{E}_i \tag{2.12}$$

where $\mathbf{1}$ is the unit tensor for 3-space.

A kinematical framework of this type is appropriate when modeling the response of thin extensible rods in which shear of the cross-section, relative to the rod axis, is suppressed. We refer to Antman [1] for comprehensive discussion. The kinematical description is completed by introducing a tensor $\mathbf{W}$ defined by:

$$\mathbf{W} = \mathbf{e}'_i \otimes \mathbf{e}_i = W_{ij}\mathbf{e}_i \otimes \mathbf{e}_j \ ; \qquad W_{ij} = \mathbf{e}_i \cdot \mathbf{e}'_j \tag{2.13}$$

This furnishes the rate of change of $\{\mathbf{e}_i\}$ with respect to S:

$$\mathbf{e}'_i = \mathbf{W}\mathbf{e}_i \tag{2.14}$$

The orthonormality of $\mathbf{e}_i$ implies that $\mathbf{W}$ is *skew* i.e. $\mathbf{W}^T = -\mathbf{W}$. Thus $\mathbf{W}(S)$ is equivalent to a vector-valued function $\mathbf{w}(S)$, in the sense that $\mathbf{W}\mathbf{u} =$

$\mathbf{w} \times \mathbf{u}$ for any vector $\mathbf{u}$. The use of $\mathbf{w}$ allows Eqn.(2.14) to be written in the form

$$\mathbf{e}'_i = \mathbf{w} \times \mathbf{e}_i \tag{2.15}$$

The relation between the components $w_i (= \mathbf{w} \cdot \mathbf{e}_i)$ and W_{ij} is well known:

$$w_i = \frac{1}{2} e_{ijk} W_{kj} \quad , \quad W_{kj} = w_i \, e_{ijk} \tag{2.16}$$

where e_{ijk} is the usual permutation symbol ($e_{123} = 1$).

Let Ω be the skew tensor defined by

$$\Omega = \mathbf{R}^T \mathbf{W} \mathbf{R} = W_{ij} (\mathbf{E}_i \otimes \mathbf{E}_j) \tag{2.17}$$

and let $\boldsymbol{\kappa} = k_i \mathbf{E}_i$ be its vector-equivalent, then we will have

$$k_i = \frac{1}{2} e_{ijk} W_{kj} \quad , \quad W_{kj} = k_i \, e_{ijk} \tag{2.18}$$

Evidently $k_i = w_i$ and $\mathbf{w} = \mathbf{R}\boldsymbol{\kappa}$. It is easy to show that λ and $\boldsymbol{\kappa}$ are invariant under superimposed *rigid* deformation

$$\mathbf{r}(S) \Rightarrow \mathbf{Q}\mathbf{r}(S) + \mathbf{c} \ , \quad \mathbf{e}_i(S) \Rightarrow \mathbf{Q}\mathbf{e}_i(S) \ , \quad \mathbf{e}'_i(S) \Rightarrow \mathbf{Q}\mathbf{e}'_i(S) \tag{2.19}$$

where $\mathbf{Q}$ is an arbitrary fixed rotation and $\mathbf{c}$ is an arbitrary fixed vector. Because of this invariance property, it is natural to formulate a theory for elastic rods by introducing a strain energy, w, per unit length of the reference placement, that depends on $\lambda, \boldsymbol{\kappa}$ and $\boldsymbol{\kappa}^0$

$$w = w(\lambda, \boldsymbol{\kappa}, \boldsymbol{\kappa}^0) \tag{2.20}$$

where $k_i^0 = \frac{1}{2} e_{ijk} \mathbf{E}_k \cdot \mathbf{E}'_j$ which are the values of $k_i(S)$ in the configuration $\{\mathbf{X}, \mathbf{E}_i\}$. We will leave any dependence on arclength S that arise due to nonuniformity of the material properties, or the presence of non-zero values of functions $k_i^0(S)$ tacit, and we just consider $w = w(\lambda, \boldsymbol{\kappa})$, as it does not affect the considerations of this work.

A model of elastic cables may be obtained by eliminating k_i(and k_i^0) from the list of arguments of the strain energy function. Alternatively, if dependence on λ is eliminated (i.e. by setting $\lambda = 1$), we recover the theory of inextensible rods considered in [36]. Retention of the full list enables us to consider a wide range of problems using a single theory.

2.2 The Variation of the Potential Energy

The total potential energy of the rod is the sum of strain energy, the potential energy of external loading like Dead-load end ($\mathbf{f_L}$,$\mathbf{f_0}$) and the distributed forces ($\mathbf{b}$).

$$E = \int_0^L (w(\lambda, k_i) - \mathbf{b}\cdot\mathbf{r})dS - (\mathbf{f}_L\cdot\mathbf{r}(L) + \mathbf{f}_0\cdot\mathbf{r}(0)) \qquad (2.21)$$

We minimize E to obtain the equilibrium configuration.

Let us consider a family of kinematically admissible configurations $\{\mathbf{r}^*(S;\varepsilon), \mathbf{e}_i^*(S;\varepsilon)\}$ with $\{\mathbf{r}^*(S;0), \mathbf{e}_i^*(S;0)\}$=$\{\mathbf{r}(S), \mathbf{e}_i(S)\}$ an Energy-minimizing configuration. Here kinematic admissibility means that, for each fixed ε , $\mathbf{r}^*(S;\varepsilon)$ and $\mathbf{e}_i^*(S;\varepsilon)$ are at least piecewise C^2 on $[0, L]$ and satisfy (2.2);

$$\mathbf{r}'^* = \lambda^*\, \mathbf{e}_1^* \;; \quad \lambda^* = |\mathbf{r}^{*\prime}| \qquad (2.22)$$

Let us just consider the strain energy of the rod which is the functional of the configuration $\{\mathbf{r}^*, \mathbf{e}_i^*\}$

$$E_s = \int_0^L w(\lambda, k)dS \qquad (2.23)$$

Then the first and the second variations of E_s at the configuration $\{\mathbf{r}, \mathbf{e}_i\}$ are

$$\overset{\circ}{E}_s = \int_0^L [(\frac{\partial w}{\partial \lambda})\overset{\circ}{\lambda} + (\frac{\partial w}{\partial k_i})\overset{\circ}{k}_i]dS \qquad (2.24)$$

and

$$\overset{\circ\circ}{E}_s = \int_0^L [(\frac{\partial^2 w}{\partial^2 \lambda})(\overset{\circ}{\lambda})^2 + (\frac{\partial w}{\partial \lambda})\overset{\circ\circ}{\lambda} + 2D_i\overset{\circ}{k}_i\overset{\circ}{\lambda} + C_{ij}\overset{\circ}{k}_i\overset{\circ}{k}_j + M_i\overset{\circ\circ}{k}_i]dS \tag{2.25}$$

respectively, where

$$M_i = \frac{\partial w}{\partial k_i}\ ,\ D_i = \frac{\partial^2 w}{\partial \lambda \partial k_i}\ ,\ C_{ij} = \frac{\partial^2 w}{\partial k_i \partial k_j}\ . \tag{2.26}$$

Also we define

$$\mathbf{M} = M_i\mathbf{e}_i,\quad \mathbf{D} = D_i\mathbf{e}_i,\quad \mathbf{C} = C_{ij}\mathbf{e}_i \otimes \mathbf{e}_j \tag{2.27}$$

In the next section we interpret **M** as the moment exerted on the part $[0, S)$ of the rod by the remainder $[S, L]$.

Henceforth in this chapter, ($\circ$) denotes derivative of function with respect to ε , evaluated at ε=0. The $\{\overset{\circ}{\lambda}, \overset{\circ}{k}_i\}$ and $\{\overset{\circ\circ}{\lambda}, \overset{\circ\circ}{k}_i\}$ are the first and second variations of $\{\lambda, k_i\}$ induced by the variations of $\{\mathbf{r}^*, \mathbf{e}_i^*\}$. To analyze the structure of these variations near ε=0 , we write

$$\mathbf{e}^*{}_i(S,\varepsilon) = \mathbf{e}_i(S) + \varepsilon\overset{\circ}{\mathbf{e}}_i(S) + \frac{1}{2}\varepsilon^2\overset{\circ\circ}{\mathbf{e}}_i(S) + o(\varepsilon^2) \tag{2.28}$$

$$\mathbf{r}^*(S,\varepsilon) = \mathbf{r}(S) + \varepsilon\mathbf{u}(S) + \frac{1}{2}\varepsilon^2\mathbf{v}(S) + o(\varepsilon^2) \tag{2.29}$$

where $\mathbf{u} = \overset{\circ}{\mathbf{r}}$ and $\mathbf{v} = \overset{\circ\circ}{\mathbf{r}}$.

Since $\{\mathbf{e}_i^*(S;\varepsilon)\}$ is an orthonormal basis for all $S \in [0, L]$ and $\varepsilon \in (-\varepsilon_0, \varepsilon_0)$,
i.e. in every kinematically admissible configuration as Eqn.(2.11) we have

$$\mathbf{e}^*{}_i(S,\varepsilon) = \mathbf{R}^*(S,\varepsilon)\mathbf{E}_i(S) \tag{2.30}$$

where $\mathbf{R}^*(S,\varepsilon)$ is a rotation. Then

$$\frac{d}{d\varepsilon}\mathbf{e}^*{}_i = \left(\frac{d}{d\varepsilon}\mathbf{R}^*\right)(\mathbf{R}^*)^T\mathbf{e}^*{}_i = \boldsymbol{\alpha}^*\mathbf{e}^*{}_i \tag{2.31}$$

where $\boldsymbol{\alpha}^*(S,\varepsilon)$ is a skew matrix. Then introducing $\mathbf{a}^*(S,\varepsilon)$ as its vector-equivalent we have

$$\frac{d}{d\varepsilon}\mathbf{e}^*{}_i = \mathbf{a}^* \times \mathbf{e}^*{}_i \tag{2.32}$$

Let $\mathbf{a}(S) = \mathbf{a}^*(S,0)$, then we have

$$\overset{\circ}{\mathbf{e}}_i = \mathbf{a} \times \mathbf{e}_i \tag{2.33}$$

Further differentiation of Eqn.(2.32) yields

$$\overset{\circ\circ}{\mathbf{e}}_i = \mathbf{a} \times (\mathbf{a} \times \mathbf{e}_i) + \mathbf{b} \times \mathbf{e}_i \tag{2.34}$$

where $\mathbf{b}(S) = \frac{d\mathbf{a}^*}{d\varepsilon}|_{\varepsilon=0}$. Also we have

$$\lambda^*(S,\varepsilon) = \lambda(S) + \varepsilon\overset{\circ}{\lambda}(S) + \frac{1}{2}\varepsilon^2\overset{\circ\circ}{\lambda}(S) + o(\varepsilon^2) \tag{2.35}$$

On combining this with (2.28), (2.29) and the constraint (2.22), we obtain the compatibility conditions

$$\mathbf{u}'(S) = \overset{\circ}{\lambda}\mathbf{e}_1 + \lambda\mathbf{a} \times \mathbf{e}_1 = \overset{\circ}{\lambda}\mathbf{e}_1 + \mathbf{a} \times \mathbf{r}' \tag{2.36}$$

and

$$\mathbf{v}'(S) = 2\overset{\circ}{\lambda}(\lambda\mathbf{a} \times \mathbf{e}_1) + \mathbf{a} \times (\mathbf{a} \times \mathbf{r}') + \overset{\circ\circ}{\lambda}\mathbf{e}_1 + \mathbf{b} \times \mathbf{r}' \tag{2.37}$$

Formulae for the first and second variations of $\kappa_i^*(S,\varepsilon)$ may be derived from (2.13) and (2.18):

$$\overset{\circ}{\kappa}_i = \frac{1}{2}e_{ijk}(\overset{\circ}{\mathbf{e}}_k \cdot \mathbf{e}'_j + \mathbf{e}_k \cdot \overset{\circ}{\mathbf{e}'}_j) \tag{2.38}$$

Using (2.33) gives

$$\overset{\circ}{\kappa}_i = \frac{1}{2}e_{ijk}[\mathbf{a} \times \mathbf{e}_k \cdot \mathbf{e}'_j + \mathbf{e}_k \cdot (\mathbf{a}' \times \mathbf{e}_j + \mathbf{a} \times \mathbf{e}'_j)] \tag{2.39}$$

The terms involving a cancel, and we obtain

$$\overset{\circ}{\kappa}_i = \frac{1}{2} e_{ijk} \mathbf{a}' \cdot \mathbf{e}_j \times \mathbf{e}_k = \mathbf{a}' \cdot (\frac{1}{2} e_{ijk} e_{jkm} \mathbf{e}_m) \tag{2.40}$$

One of the $e - \delta$ identies furnishes the result

$$\overset{\circ}{\kappa}_i = \mathbf{e}_i \cdot \mathbf{a}' \tag{2.41}$$

With the use of (2.34), we may also show that

$$\overset{\circ\circ}{\kappa}_i = \mathbf{e}_i \cdot \mathbf{a}' \times \mathbf{a} + \mathbf{e}_i \cdot \mathbf{b}' \tag{2.42}$$

2.3 Lattices

Consider a network consisting of n rods, the jth rod has the arclength L_j in its reference configuration. These rods are connected at l unconstrained nodes located at the positions $\mathbf{x}_k; k = 1, ..., l$. After deformation, the nodes displaced to the (unknown) positions $\mathbf{y}_k$. Let the set of all unconstrained nodes be denoted by K. We let m nodes, belonging to the set H, be fixed at prescribed positions $\mathbf{z}_h; h = 1, ..., m$. At each of the unconstrained nodes, a dead load $\mathbf{q}_k, k \in K$ is applied. The collection of all node labels is $K \bigcup H$. The deformed configuration of the jth rod is described by the vector-valued position function $\mathbf{r}_j(S_j)$ and the orientation traid $\{\mathbf{e}_i(S_j)\}_j$ where $S_j \in [0, L_j]$ is the reference arclength along the jth rod. Following Cannarozzi [6], we introduce the sets

$$\begin{aligned} I_k &= \{j : S_j = 0 \quad \text{at node } k \in K\} \\ E_k &= \{j : S_j = L_j \quad \text{at node } k \in K\} \end{aligned} \tag{2.43}$$

and

$$\begin{aligned} I_h &= \{j : S_j = 0 \quad \text{at node } h \in H\} \\ E_h &= \{j : S_j = L_j \quad \text{at node } h \in H\} \end{aligned} \tag{2.44}$$

Henceforth we use superscripts 0 and L to denote the values of functions at $S_j = 0$ and $S_j = L_j$ respectively. Thus

$$\mathbf{r}_j^0 = \mathbf{r}_j(0) \quad \text{and} \quad \mathbf{r}_j^L = \mathbf{r}_j(L_j) \tag{2.45}$$

These are subjected to the following continuity constraints

$$\begin{cases} \mathbf{r}_j^0 = \mathbf{y}_k, & j \in I_k \\ \mathbf{r}_j^L = \mathbf{y}_k, & j \in E_k \end{cases} \tag{2.46}$$

and

$$\begin{cases} \mathbf{r}_j^0 = \mathbf{z}_h, & j \in I_h \\ \mathbf{r}_j^L = \mathbf{z}_h, & j \in E_h \end{cases} \tag{2.47}$$

The potential energy, E, of a configuration of the entire lattice, suppressing the distributed forces, is

$$E = \sum_{j=1}^{n} E_{sj} - \sum_{k=1}^{l} \mathbf{q}_k \cdot \mathbf{y}_k \tag{2.48}$$

where E_{sj} is the total strain energy of the jth rod. A configuration of the lattice is equilibrated if and only if the associated first variation of the energy vanishes for all admissible variation of the kinematical variables:

$$0 = \sum_{j=1}^{n} \overset{\circ}{E}_{sj} - \sum_{k=1}^{l} \mathbf{q}_k . \mathbf{u}_k ; \quad \mathbf{u}_k = \overset{\circ}{\mathbf{y}}_k \tag{2.49}$$

where using Eqns.(2.24),(2.26),(2.28),(2.36) and (2.41) gives

$$\overset{\circ}{E}_{sj} = \int_0^{L_j} [(\frac{\partial w}{\partial \lambda})\mathbf{e}_1 \cdot \mathbf{u}' + \mathbf{M} \cdot \mathbf{a}']dS \tag{2.50}$$

in which the index j has been suppressed in the integrand for the sake of clarity.

To proceed it is necessary to account explicity for the fact that the variation $\mathbf{u}(S)$ and $\mathbf{a}(S)$ are not independent. In particular, scalar multiplication of (2.36) with $\mathbf{e}_\alpha(\alpha = 2, 3)$ furnishes the constraints:

$$\mathbf{u}' \cdot \mathbf{e}_\alpha + \mathbf{r}' \cdot \mathbf{a} \times \mathbf{e}_\alpha = 0; \quad \alpha = 2, 3 \tag{2.51}$$

Using this we have $\overset{\circ}{E}_{s_j} = T_j$, where

$$T_j = \int_0^{L_j} [(\frac{\partial w}{\partial \lambda})\mathbf{e}_1 \cdot \mathbf{u}' + \mathbf{M} \cdot \mathbf{a}' + F_\alpha(\mathbf{u}' \cdot \mathbf{e}_\alpha + \mathbf{r}' \cdot \mathbf{a} \times \mathbf{e}_\alpha)]dS \tag{2.52}$$

or

$$T_j = \int_0^{L_j} [\mathbf{F} \cdot \mathbf{u}' + \mathbf{M} \cdot \mathbf{a}' + \mathbf{r}' \cdot \mathbf{a} \times (F_\alpha \mathbf{e}_\alpha)]dS \tag{2.53}$$

where $F_\alpha(S), \alpha \in \{2, 3\}$, are Lagrange multipliers, and

$$\mathbf{F} = (\frac{\partial w}{\partial \lambda})\mathbf{e}_1 + F_\alpha \mathbf{e}_\alpha \tag{2.54}$$

Since $\mathbf{r}'$ is parallel to $\mathbf{e}_1$, the last term in Eqn.(2.53) can be replaced with $\mathbf{a} \cdot \mathbf{F} \times \mathbf{r}'$, and integration by parts yields

$$T_j = U_j + \mathbf{F}_j^L \cdot \mathbf{u}_j^L + \mathbf{M}_j^L \cdot \mathbf{a}_j^L - \mathbf{F}_j^0 \cdot \mathbf{u}_j^0 - \mathbf{M}_j^0 \cdot \mathbf{a}_j^0 \tag{2.55}$$

where

$$U_j = -\int_0^{L_j} [\mathbf{u} \cdot \mathbf{F}' + \mathbf{a} \cdot (\mathbf{M}' - \mathbf{F} \times \mathbf{r}')]dS \tag{2.56}$$

These expressions may be used to reduce the stationary-energy condition, Eqn.(2.49), to

$$\begin{aligned} 0 = &\sum_{j=1}^{n} U_j - \sum_{k=1}^{l} \mathbf{u}_k \cdot [\mathbf{q}_k - (\sum_{j \in E_k} \mathbf{F}_j^L - \sum_{j \in I_k} \mathbf{F}_j^0)] \\ &+ \sum_{k=1}^{l} (\sum_{j \in E_k} \mathbf{M}_j^L \cdot \mathbf{a}_j^L - \sum_{j \in I_k} \mathbf{M}_j^0 \cdot \mathbf{a}_j^0) \\ &+ \sum_{h=1}^{m} (\sum_{j \in E_h} \mathbf{M}_j^L \cdot \mathbf{a}_j^L - \sum_{j \in I_h} \mathbf{M}_j^0 \cdot \mathbf{a}_j^0) \end{aligned} \tag{2.57}$$

wherein we have imposed the constraints

$$\mathbf{u}_j^0 = \mathbf{u}_k,\ \ j \in I_k;\ \ \mathbf{u}_j^L = \mathbf{u}_k,\ \ j \in E_k;\ \ \mathbf{u}_j^0 = 0,\ \ j \in I_h;\ \ \mathbf{u}_j^L = 0,\ \ j \in E_h \tag{2.58}$$

Restrictions on the virtual rotations $\mathbf{a}_j^0$ and $\mathbf{a}_j^L$ must also be imposed in accordance with the particular type of nodal connection under consideration.

Now Eqn.(2.57) must be satisfied for all admissible $\mathbf{u}_k$, $\mathbf{a}_j^0$ and $\mathbf{a}_j^L$. Null values of these variations are admissible in all lattice types, and for this choice (2.57) requires that the sum $\sum U_j$ vanish. By choosing $\mathbf{u}(S)$ and $\mathbf{a}(S)$ to be non-zero in each of the n rods in succession, we conclude that $U_j = 0; j = 1, ..., n$, where U_j is given by (2.56). The *multiplier rule* [10] of the calculus of variations then yields the equilibrium equation in each rod:

$$\mathbf{F}' = 0,\ \ \mathbf{M}' = \mathbf{F} \times \mathbf{r}' \tag{2.59}$$

These are identical to the classical equations of rod theory [5], in the absence of distributed load, provided that $\mathbf{F}(S)$ and $\mathbf{M}(S)$ are identified with the force and moment, respectively, exerted by the part $[S, L]$ on the remainder $[0, S)$. The moment is given by the constitutive equation (2.26) with (2.27). In view of (2.54), it is only the tangential component of the force that is determined by a constitutive equation. The transverse components F_α are shear reactions that are workless in any variation of the configuration compatible with (2.51). They are determined by equilibrium considerations alone.

With (2.59) satisfied in each rod, all of the U_j in (2.57) vanish, and the remaining expression must be satisfied for all $\mathbf{u}_k$ and for all admissible $\mathbf{a}_j^0$ and $\mathbf{a}_j^L$. On setting $\mathbf{a}_j^0 = 0$, $\mathbf{a}_j^L = 0$ and taking all but one of the $\mathbf{u}_k$ to be zero in succession, we obtain the nodal force balance equations

$$\sum_{j \in E_k} \mathbf{F}_j^L - \sum_{j \in I_k} \mathbf{F}_j^0 = \mathbf{q}_k,\ \ k \in K \tag{2.60}$$

These require that the net effect of the nodal reactions (the forces exerted by the nodes on the rods) be such as to balance the applied forces.

Additional equilibrium conditions may be derived, but these vary depending on the class of nodal constraint. We present discussions of three of these classes.

(a) Unrestricted rotations

If the rod rotations are unrestricted at the nodes, then there are no kinematical constraints on virtual displacements or rotations beyond those that have already been imposed. Nodal connections of this type are appropriate in a theory of cable networks, or in models of ball and socket joints in structural lattices.

For illustrative purposes, suppose that a particular node $k' \in K$ is of this type. Then a necessary condition for (2.57) is

$$\sum_{j\in E_{k'}} \mathbf{M}_j^L \cdot \mathbf{a}_j^L - \sum_{j\in I_{k'}} \mathbf{M}_j^0 \cdot \mathbf{a}_j^0 = 0 \tag{2.61}$$

For each $j \in E_{k'} \bigcup I_{k'}$, the $\mathbf{a}_j^0$ and $\mathbf{a}_j^L$ may be specified independently, so the node is equilibrated only if it transmits no moment to any of the attached rods:

$$\mathbf{M}_j^0 = 0 \ , \ \ j \in I_{k'} \ \ ; \quad \mathbf{M}_j^L = 0 \ , \ \ j \in E_{k'} \tag{2.62}$$

(b) Rigidly constrained rotations

Once again we focus attention on a particular $k' \in K$ for the sake of illustration. Suppose the nodal connection is rigid in the sense that, for any two values of j in the set $E_{k'} \bigcup I_{k'}$, the angles formed by the triads $\{\mathbf{E}_i\}_j^0$ and $\{\mathbf{E}_i\}_j^L$ are preserved under deformation. This is equivalent to requiring that

$$\mathbf{R}_j^0 = \mathbf{R}_{k'} \ , \ \ j \in I_{k'} \ \ ; \quad \mathbf{R}_j^L = \mathbf{R}_{k'} \ , \ \ j \in E_{k'} \tag{2.63}$$

for some rotation $\mathbf{R}_{k'}$, where $\mathbf{R}^0$ and $\mathbf{R}^L$ are the endpoint values of the rod rotation $\mathbf{R}$ defined in (2.12). If such conditions are imposed in all configurations, then (2.31) may be used to derive an associated set of restrictions on the virtual endpoint rotations:

$$\mathbf{a}_j^0 = \mathbf{a}_{k'} \;,\; j \in I_{k'} \;;\quad \mathbf{a}_j^L = \mathbf{a}_{k'} \;,\; j \in E_{k'} \tag{2.64}$$

for some arbitrary vector $\mathbf{a}_{k'}$.

In the present case (2.61) remains valid, and (2.64) yields

$$\mathbf{a}_{k'} \cdot \boldsymbol{\mu}_{k'} = 0 \;,\quad \text{for all } \mathbf{a}_{k'} \tag{2.65}$$

where

$$\boldsymbol{\mu}_{k'} = \sum_{j \in E_{k'}} \mathbf{M}_j^L - \sum_{j \in I_{k'}} \mathbf{M}_j^0 \tag{2.66}$$

Thus the net moment at the node vanishes: $\boldsymbol{\mu}_{k'} = 0$

(c) Concurrent axes of rotation

As a final example, let the rods joined at node $k' \in K$ be constrained in such a way as to pivot freely about a common axis with orientation $\mathbf{N}$ in the reference configuration of the lattice. Then the endpoint rotations $\mathbf{R}^0$ and $\mathbf{R}^L$ are such that

$$\mathbf{R}_j^0\mathbf{N} = \mathbf{n} \;,\; j \in I_{k'} \;;\quad \mathbf{R}_j^L\mathbf{N} = \mathbf{n} \;,\; j \in E_{k'} \tag{2.67}$$

where $\mathbf{n}$ is the pivotal axis in the deformed lattice. We take $\mathbf{N}$ and $\mathbf{n}$ to be unit vectors without loss of generality.

The variational versions of (2.67) are

$$\mathbf{a}_j^0 \times \mathbf{n} = \boldsymbol{\beta} \;,\; j \in I_{k'} \;;\quad \mathbf{a}_j^L \times \mathbf{n} = \boldsymbol{\beta} \;,\; j \in E_{k'} \tag{2.68}$$

where

$$\boldsymbol{\beta} = \mathring{\mathbf{n}} \tag{2.69}$$

is the variation of $\mathbf{n}$. We note that $\boldsymbol{\beta} \cdot \mathbf{n} = 0$ because $\mathbf{n}$ is a unit vector. Now any vector $\mathbf{a}$ may be represented in the form

$$\mathbf{a} = (\mathbf{n} \cdot \mathbf{a})\mathbf{n} + \mathbf{n} \times (\mathbf{a} \times \mathbf{n}) \tag{2.70}$$

so that

$$\mathbf{a}_\mathbf{j}^0 = \alpha_j^0 \mathbf{n} + \mathbf{n} \times \boldsymbol{\beta} \quad \text{and} \quad \mathbf{a}_\mathbf{j}^L = \alpha_j^L \mathbf{n} + \mathbf{n} \times \boldsymbol{\beta} \tag{2.71}$$

for some scalars α_j^0, α_j^L.

Substitution of (2.71) into (2.61) yields

$$0 = \sum_{j \in E_{k'}} \alpha_j^L \mathbf{n} \cdot \mathbf{M}_j^L - \sum_{j \in I_{k'}} \alpha_j^0 \mathbf{n} \cdot \mathbf{M}_j^0 + \mathbf{n} \times \boldsymbol{\beta} \cdot \boldsymbol{\mu}_{k'} \tag{2.72}$$

where $\boldsymbol{\mu}_{k'}$ is given by (2.66). This must hold for arbitrary α_j^0, α_j^L, and for any $\boldsymbol{\beta}$ perpendicular to $\mathbf{n}$. By setting $\boldsymbol{\beta} = 0$ and all but one of the α's equal to zero in succession, we derive the necessary conditions

$$\mathbf{n} \cdot \mathbf{M}_j^0 = 0\ , \quad j \in I_{k'} \quad ; \quad \mathbf{n} \cdot \mathbf{M}_j^L = 0\ , \quad j \in E_{k'} \tag{2.73}$$

Thus the node transmits no *axial* moment to any of the attached rods. Finally, the last term in (2.72) requires that $\boldsymbol{\mu}_{k'}$ be orthogonal to any vector in the plane perpendicular to $\mathbf{n}$, so that $\boldsymbol{\mu}_{k'} = (\mathbf{n} \cdot \boldsymbol{\mu}_{k'})\mathbf{n}$. But

$$\mathbf{n} \cdot \boldsymbol{\mu}_{k'} = \sum_{j \in E_{k'}} \mathbf{n} \cdot \mathbf{M}_j^L - \sum_{j \in I_{k'}} \mathbf{n} \cdot \mathbf{M}_j^0 \tag{2.74}$$

and this vanishes by (2.73). Thus there is no net moment at the node:

$$\boldsymbol{\mu}_{k'} = 0 \tag{2.75}$$

Considering Examples (2-4) and (2-5) shows that a concurrent axes of rotation node behaves like a rigid node for the pure out of plane deformation and like a pinned node for the in plane deformation, which is reasonable.

2.4 Equilibrium Equations for Small Deformation

Consider initially straight rods, for the isotropic material and circular cross-section the strain energy, w, per unit length of the reference placement, can be written as

$$w(\lambda, \boldsymbol{\kappa}) = \frac{1}{2}EI(\kappa_2^2 + \kappa_3^2) + \frac{1}{2}GJ\kappa_1^2 + \frac{1}{2}EA(\lambda - 1)^2 \tag{2.76}$$

where EI and GJ are flexural and torsional rigidity. The last part of this equation is due to the extensibility of the rod. Using Eqns.(2.27), (2.26) and (2.76) gives

$$\mathbf{M} = EI(\kappa_2\mathbf{e}_2 + \kappa_3\mathbf{e}_3) + GJ\kappa_1\mathbf{e}_1 \tag{2.77}$$

Using Eqn.(2.15) and since $k_i = w_i$ we get

$$\begin{aligned} \mathbf{e}'_1 &= \kappa_3\mathbf{e}_2 - \kappa_2\mathbf{e}_3 \\ \mathbf{e}'_2 &= \kappa_1\mathbf{e}_3 - \kappa_3\mathbf{e}_1 \\ \mathbf{e}'_3 &= \kappa_2\mathbf{e}_1 - \kappa_1\mathbf{e}_2 \end{aligned} \tag{2.78}$$

Using Eqns.(2.78) and (2.77) gives the constitutive equation as

$$\mathbf{M} = EI\mathbf{e}_1 \times \mathbf{e}'_1 + GJ\kappa_1\mathbf{e}_1 \tag{2.79}$$

Also using (2.54) and (2.76) gives

$$\mathbf{F} = EA(\lambda - 1)\mathbf{e}_1 + F_\alpha\mathbf{e}_\alpha \ ; \quad \alpha = 2, 3 \tag{2.80}$$

Suppose the deformed curvatures and twist are not too large, and let $(\mathbf{r}^*, \mathbf{e}_i^*)$ be a one-parameter (η) family of equilibrium configurations. So we have

$$\begin{aligned} \mathbf{e}_i^* &= \mathbf{e}_i + \eta\hat{\mathbf{e}}_i + o(\eta) \\ \mathbf{r}^* &= \mathbf{r} + \eta\mathbf{u} + o(\eta) \end{aligned} \tag{2.81}$$

Associated with these, there will be small changes in **F** and **M** as

$$\begin{aligned}\mathbf{F}^* &= \mathbf{F} + \eta\hat{\mathbf{F}} + o(\eta)\\ \mathbf{M}^* &= \mathbf{M} + \eta\hat{\mathbf{M}} + o(\eta)\end{aligned} \tag{2.82}$$

Equilibrium of the stared-configuration implies that

$$(\mathbf{F}^*)' = 0, \quad (\mathbf{M}^*)' = \mathbf{F}^* \times (\mathbf{r}^*)' \tag{2.83}$$

and also we have

$$\mathbf{F}' = 0, \quad \mathbf{M}' = \mathbf{F} \times \mathbf{r}' \tag{2.84}$$

Substituting (2.82) into (2.83) with (2.84) and dividing the result by η and let $\eta \Rightarrow 0$ we get

$$\hat{\mathbf{F}}' = 0, \quad \hat{\mathbf{M}}' = \hat{\mathbf{F}} \times \mathbf{r}' + \mathbf{F} \times \hat{\mathbf{r}}' \tag{2.85}$$

We could also get these equations with taking the derivatives of equilibrium equations (2.59) with respect to η at $\eta = 0$.

Taking derivatives of equations (2.79) and (2.80) with respect to η at $\eta = 0$ gives the linearizations

$$\hat{\mathbf{M}} = EI(\hat{\mathbf{e}}_1 \times \mathbf{e'}_1 + \mathbf{e}_1 \times \hat{\mathbf{e'}}_1) + GJ(\hat{\kappa}_1\mathbf{e}_1 + \kappa_1\hat{\mathbf{e}}_1) \tag{2.86}$$

$$\hat{\mathbf{F}} = EA\hat{\lambda}\mathbf{e}_1 + EA(\lambda - 1)\hat{\mathbf{e}}_1 + F_\alpha\hat{\mathbf{e}}_\alpha + \hat{F}_\alpha\mathbf{e}_\alpha \tag{2.87}$$

Since at the undeformed configuration before applying the load we have $\mathbf{F} = 0$, $\boldsymbol{\kappa} = 0$ and $\lambda = 1$ so

$$\mathbf{e'}_i = 0, \qquad \mathbf{r}' = \mathbf{e}_1 \tag{2.88}$$

Using these Eqns.(2.85), (2.86) and (2.87) can be simplified as

$$\hat{\mathbf{F}}' = 0, \quad \hat{\mathbf{M}}' = \hat{\mathbf{F}} \times \mathbf{e}_1 \tag{2.89}$$

$$\hat{\mathbf{M}} = EI(\mathbf{e}_1 \times \hat{\mathbf{e}}_1') + GJ\hat{\kappa}_1\mathbf{e}_1 \tag{2.90}$$

$$\hat{\mathbf{F}} = EA\hat{\lambda}\mathbf{e}_1 + \hat{F}_\alpha\mathbf{e}_\alpha \tag{2.91}$$

Taking derivative of Eqn.(2.90) with respect to arclength S gives

$$\hat{\mathbf{M}}' = EI(\mathbf{e}_1 \times \hat{\mathbf{e}}_1'') + GJ\hat{\kappa}_1'\mathbf{e}_1 \tag{2.92}$$

Considering Eqn.(2.81) gives

$$\hat{\mathbf{e}}_1 = \mathbf{u}' \tag{2.93}$$

Taking derivative of equation $\mathbf{u} = u_i\mathbf{e}_i$ with respect to S and using (2.88) gives

$$\mathbf{u}' = u_i'\mathbf{e}_i \tag{2.94}$$

Using Eqns.(2.93),(2.94) with (2.92) gives

$$\hat{\mathbf{M}}' = EI(\mathbf{e}_1 \times u_i'''\mathbf{e}_i) + GJ\hat{\kappa}_1'\mathbf{e}_1 \tag{2.95}$$

After applying the nodal force $\mathbf{P} = p_i\mathbf{e}_i$ we have

$$\hat{\mathbf{F}} = \mathbf{P} = p_i\mathbf{e}_i \tag{2.96}$$

Considering (2.95) with the equation (2.89) for the moment with (2.96) gives

$$EIu_\alpha''' = -p_\alpha\ , \quad \alpha = 2,3 \qquad GJ\hat{\kappa}_1' = 0 \tag{2.97}$$

Also considering (2.96) with (2.91) gives

$$EA\hat{\lambda} = p_1, \quad \hat{F}_\alpha = p_\alpha\ , \quad \alpha = 2,3 \tag{2.98}$$

Using results analogous to Eqn.(2.36) gives the compatibility conditions as

$$u_1' = \hat{\lambda}, \quad u_2' = \lambda a_3, \quad u_3' = -\lambda a_2 \tag{2.99}$$

where a_2 and a_3 are the infinitesimal cross-section rotations. With $\lambda = 1$ we get

$$u_1' = \hat{\lambda}, \quad u_2' = a_3, \quad u_3' = -a_2 \tag{2.100}$$

Using results analogous to Eqn.(2.41) gives $\hat{\kappa}_1 = a_1'$, so applying this with (2.97) gives

$$EIu_\alpha'''' = -p_\alpha \ , \quad \alpha = 2,3 \qquad GJa_1'' = 0 \tag{2.101}$$

Equations (2.98) and (2.101), are the linear equations of equilibrium for the small deformation. So equations (2.98), (2.101) and (2.100) give the deformation of a lattice for the small deformation. These equations have been applied for different types of lattices with different kind of boundary conditions which were mentioned in section (2.1) employing the FDM and FEM. The Gauss-Elimination method was used to solve the equations of equilibrium. The following section show some results of some of them.

2.5 Numerical Results

Example (2-1): The space truss shown in Figure (2.1a) was also considered in [34] (page 207). In the initial configuration nodes 5, 6, 7, 12, 13, 14,19, 20, and 21 are in plane $z = 2.5\ ft$; all other nodes have $z = 0$. Nodes 1, 4, 22, and 25 are restrained in the x, y, and z directions. Nodes 2, 3, 8, 11, 15, 18, 23, and 24 have applied loads of $P_z = 2.5\ klb$; nodes 9, 10, 16 and 17 have $P_z = 5\ klb$. All elements have a cross-section 1 in^2 and $E = 29000\ kpsi$. The deformed configuration shown in Figure (2.1b) is scaled here. For the displacement u_z at nodes 2, 5, 6, 9, 13, our results are $0.1339'', 0.10523'', 0.1508'', 0.1679'', 0.1679''$, whereas the results given in [34] are $0.134'', 0.105'', 0.150'', 0.168'', 0.168''$.

Example (2-2): Here we consider the frame with the initial configuration shown in Figure (2.2a) in the plane $z = 0$. The applied out of plane loads are $F_1 = 5$ *klb* and $F_2 = 1$ *klb*. The material properties are $E = 29000$ *kpsi*, G=20000 *kpsi* and the radius of the circular cross section is 1 *in*. The displacements in the deformed configuration are $u_z = 2.72$ *in* for nodes 9, 10, 16, 17 and $u_z = 3.561$ *in* for node 13. The scaled deformed configuration is shown in Figure (2.2b).

Example (2-3): Space truss shown in Figure (2.3b) is also considered in [9] (page 165). Figure (2.3a) shows two views of the space truss. The rods have a cross-sectional area of 10 cm^2, except for rods AB and BC which have an area of 20 cm^2 and also $E = 200\ GNm^{-2}$ is for all rods. Our results for displacements of node A are $u_x = -0.5469\ mm$, $u_y = 3.12601\ mm$, $u_z = 0$ and for node B are $u_x = -0.5103\ mm$, $u_y = 1.3418\ mm$, $u_z = 0$.The results given in [9] for displacements of node A are $u_x = -0.5476\ mm$, $u_y = 3.09846\ mm$, $u_z = 0$ and for node B are $u_x = -0.5\ mm$, $u_y = 1.34375\ mm$, $u_z = 0$. The scaled deformed configuration is also shown in Figure (2.3b).

Example (2-4): The out of plane deformation for different boundary conditions is compared here. From the scaled deformed configurations shown in Figure (2.4) we see that the out of plane deformation for concurrent axes-jointed node is the same as rigid-jointed node.

Example (2-5): Similar to the previous example the in-plane deformation for different boundary conditions is compared here. From the scaled deformed configurations shown in Figure (2.5) we see that the out of plane deformation for concurrent axes-jointed node is the same as pinned-jointed node.

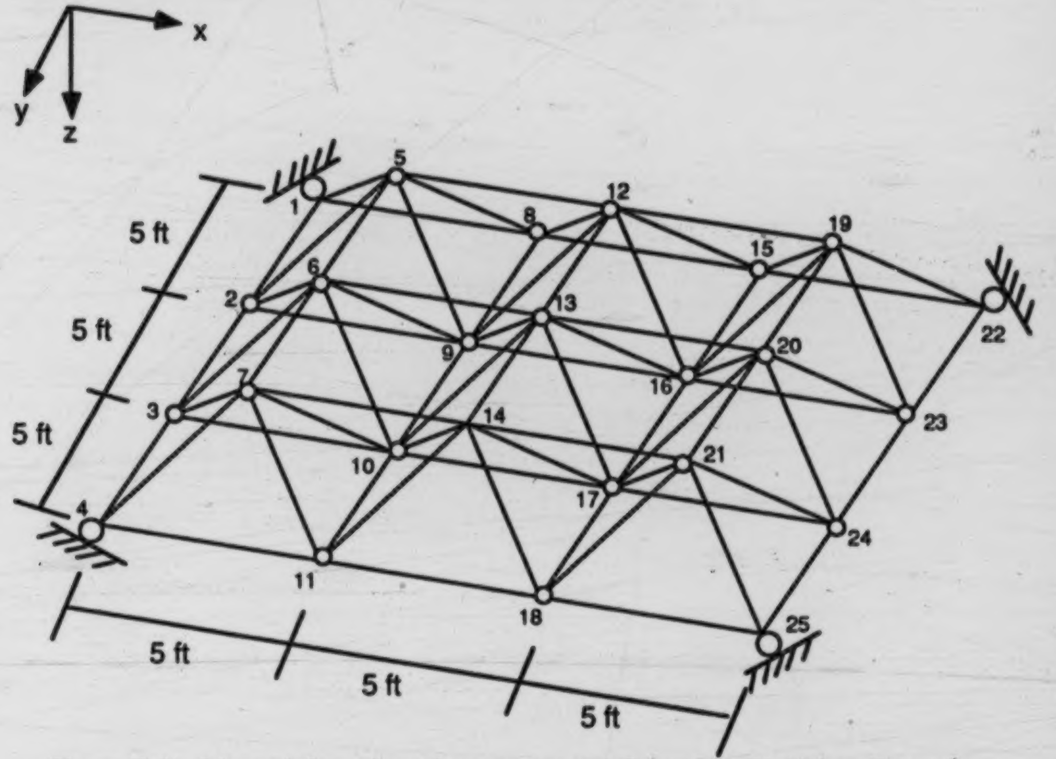
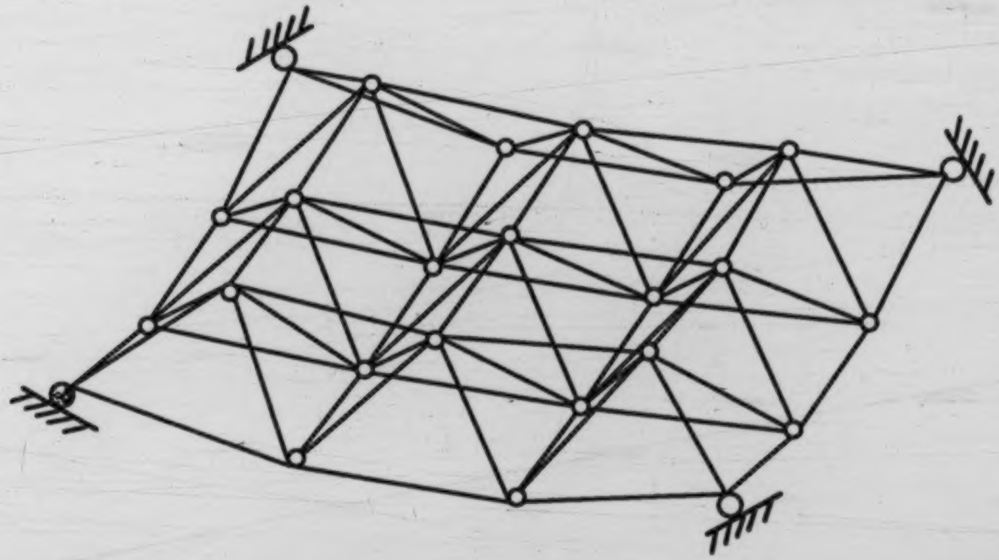

x
y
z
5 ft
5 ft
5 ft
5 ft
5 ft
5 ft
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

Figure (2-1): Linear deformation of space truss. (a) Geometry of initial configuration



(b) Scaled deformed configuration

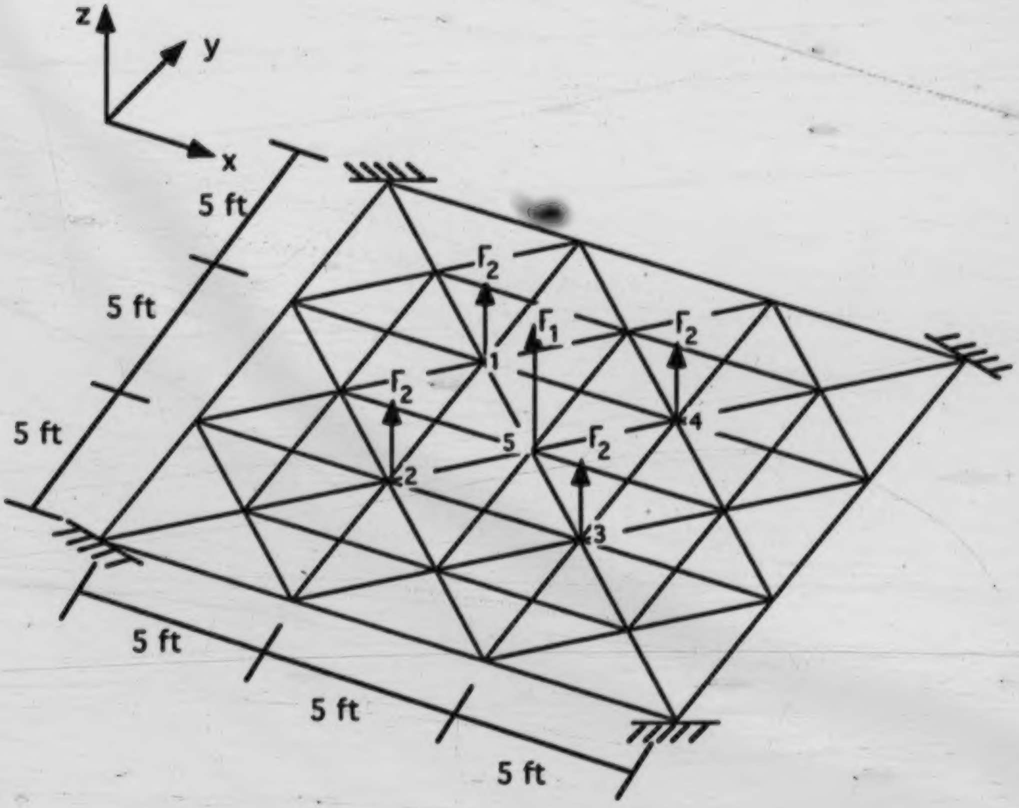
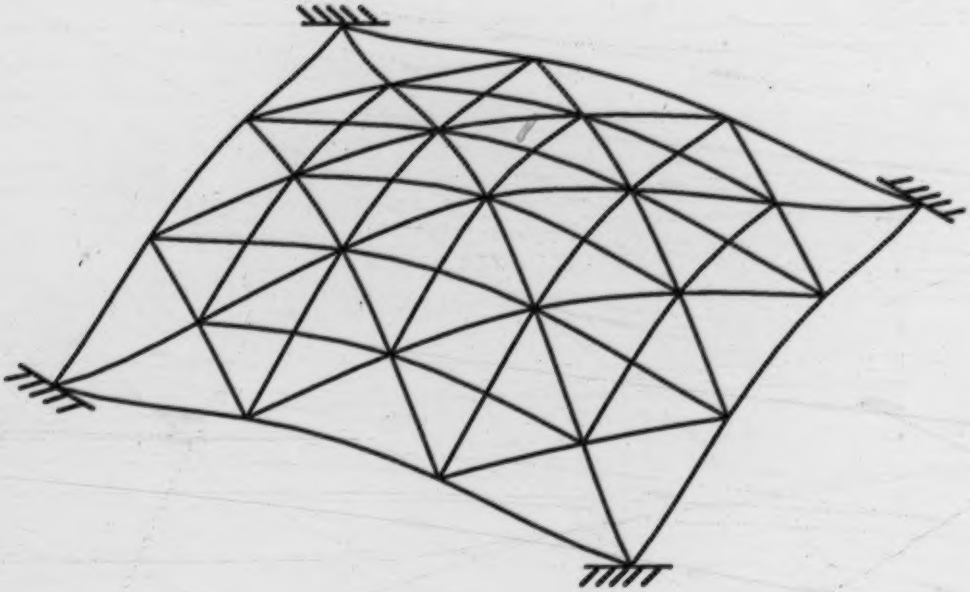

z
y
x
5 ft
5 ft
5 ft
5 ft
5 ft
5 ft
Γ_2
Γ_1
Γ_2
Γ_2
Γ_2
1
2
3
4
5

Figure (2-2): Linear out of plane deformation for plane frame.

(a) Geometry of initial configuration



(b) Scaled deformed configuration.

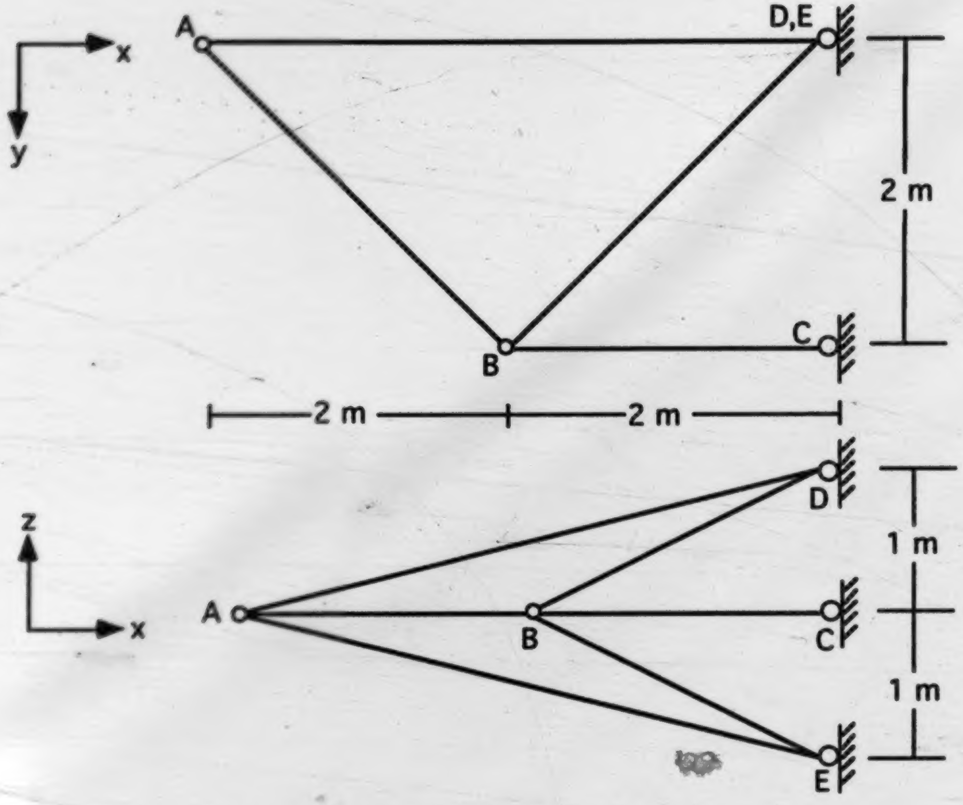
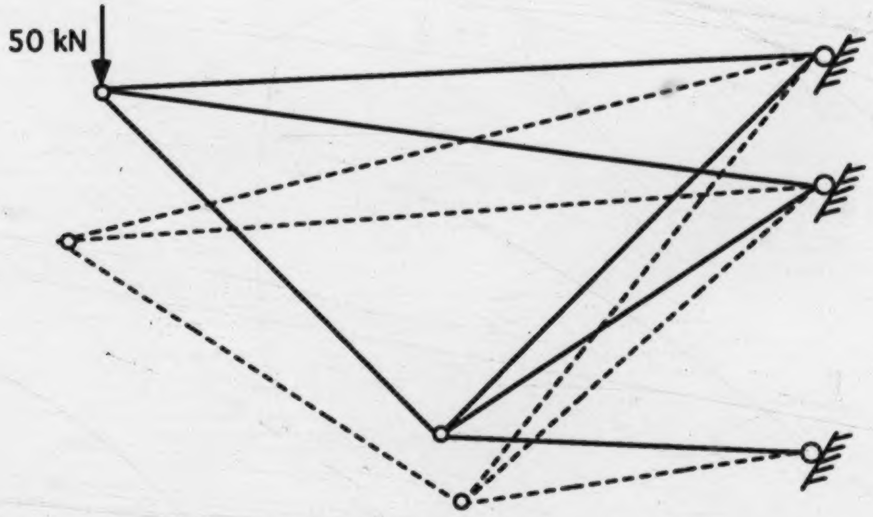

x
y
A
D,E
2 m
B
C
2 m
2 m
D
1 m
z
x
A
B
C
1 m
E

Figure (2-3): Linear deformation for space truss.

(a) The elevation and plan views of the space structure.



(b): Initial configuration and scaled deformed configuration.

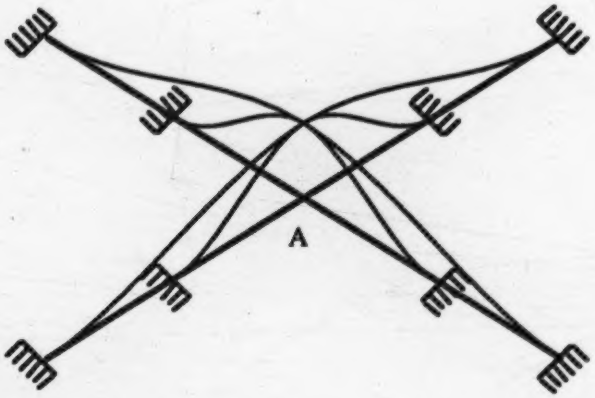
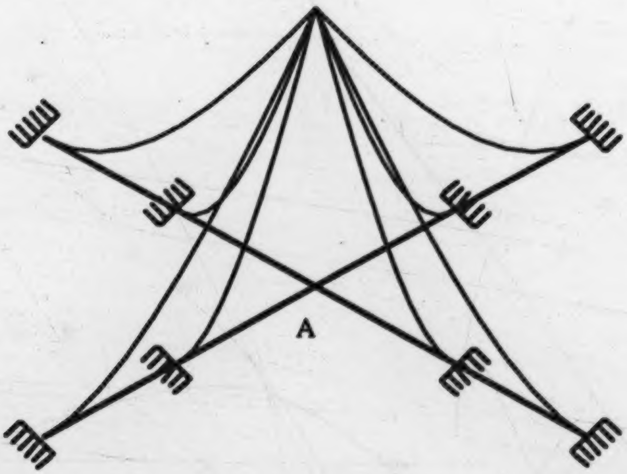
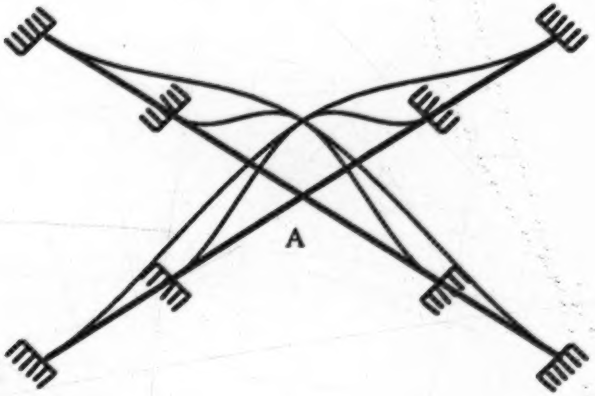


Figure (2-4): Comparison of different boundary conditions for out of plane deformation. Initial configuration (thick line) and scaled deformed configuration (thin line) for (a) Node A rigid-jointed



(b) Node A ball-jointed



(c) Node A concurrent axes-jointed.

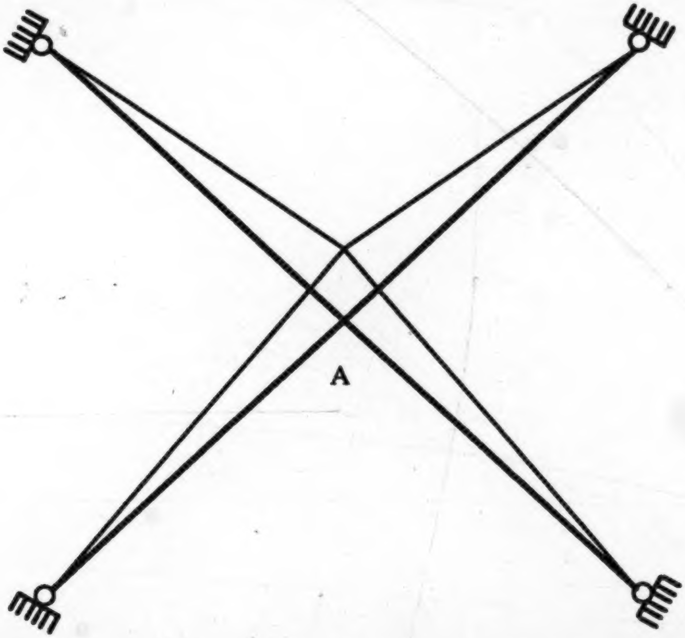
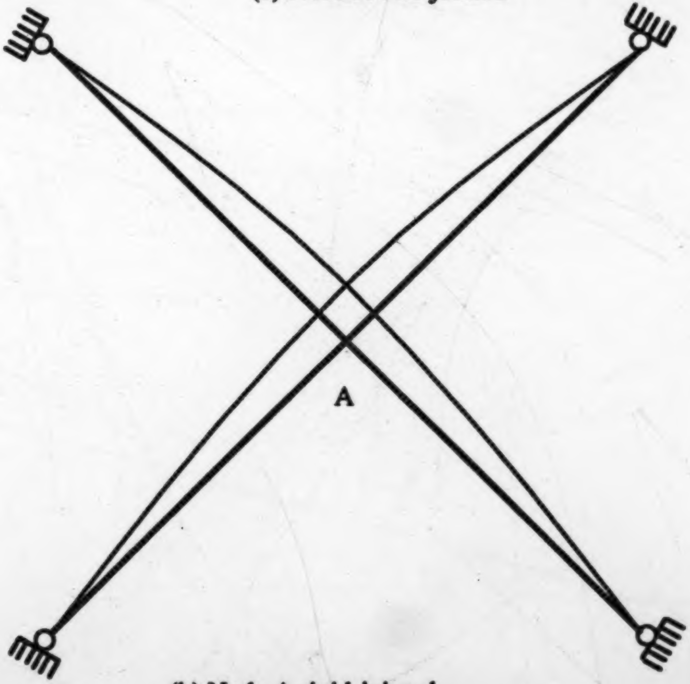
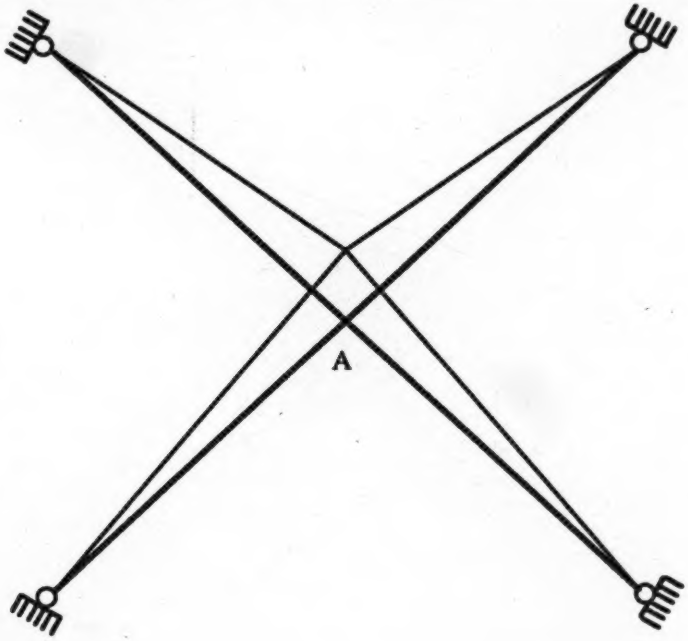


Figure (2-5): Comparison of different boundary conditions for in plane deformation. Initial configuration (thick line) and scaled deformed configuration (thin line) for
(a) Node A ball-jointed



(b) Node A rigid-jointed



(c) Node A concurrent axes-jointed.

Example (2-6): The plane frame shown in Figure (2.6a) is also considered in [9] (page 164). For the members 2-1-4 $EI_y = 15$ MNm^2 and $GI = 4$ MNm^2. For members 1-3 and 1-5, $EI_y = 8$ MNm^2 and $GI = 2$ MNm^2. Halfway along member $1-3$ a load of 100 kN is applied in the z direction. Our result for the displacement u_z at node 1 is 4.69 mm, whereas the result given in [9] is 4.69186 mm. The scaled deformed configuration is shown in Figure (2.6b).

Example (2-7): The plane frame shown in Figure (2.7a) is also considered in [9] (page 142). For the members 2-5 and 3-6 the flexural and torsional rigidities are $EI_y = 6.4$ MNm^2 and $GI = 4$ MNm^2. For all other members $EI_y = 12.5$ MNm^2 and $GI = 10$ MNm^2. Equal loads of magnitude $50kN$ are applied at points 5 and 6 in the direction normal to the plane of frame, i.e. in the z direction. Our result for the displacement u_z at nodes 5, 6 is 19.947 mm, whereas the result given in [9] is 20.011 mm. The scaled deformed configuration is shown in Figure (2.7b).

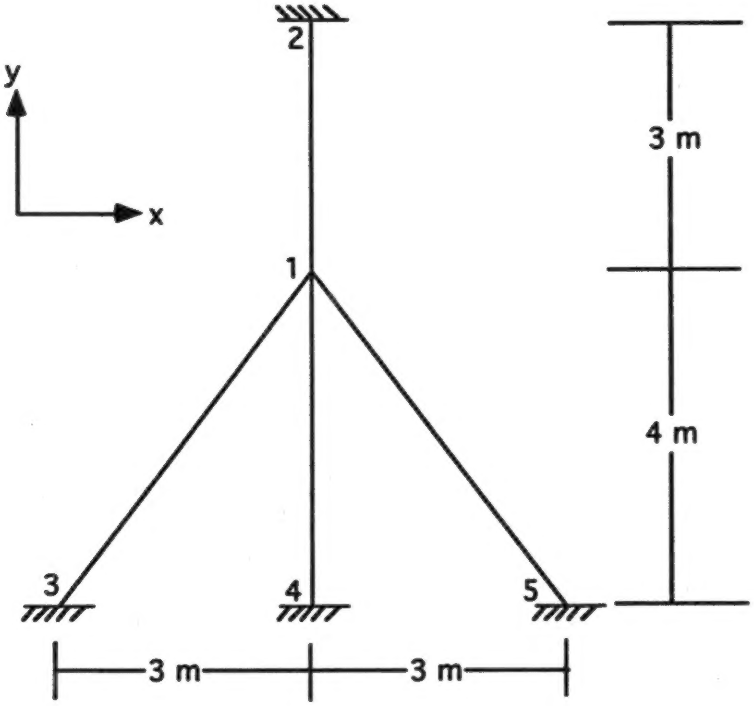
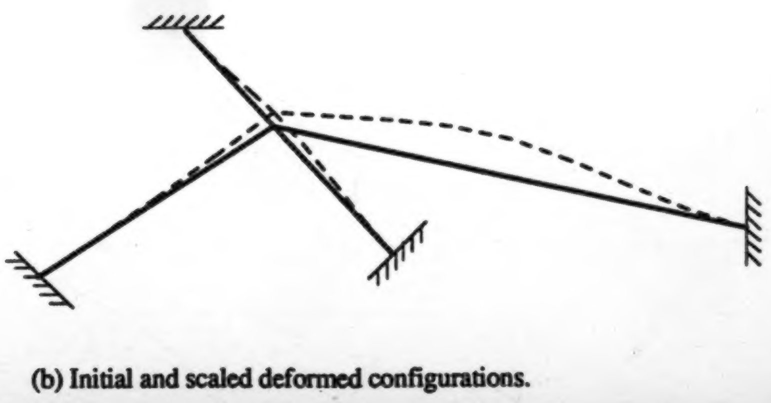


Figure (2-6): Linear out of plane deformation for plane frame with the applied load in the halfway member 1-3.
(a) Geometry of initial configuration .



(b) Initial and scaled deformed configurations.

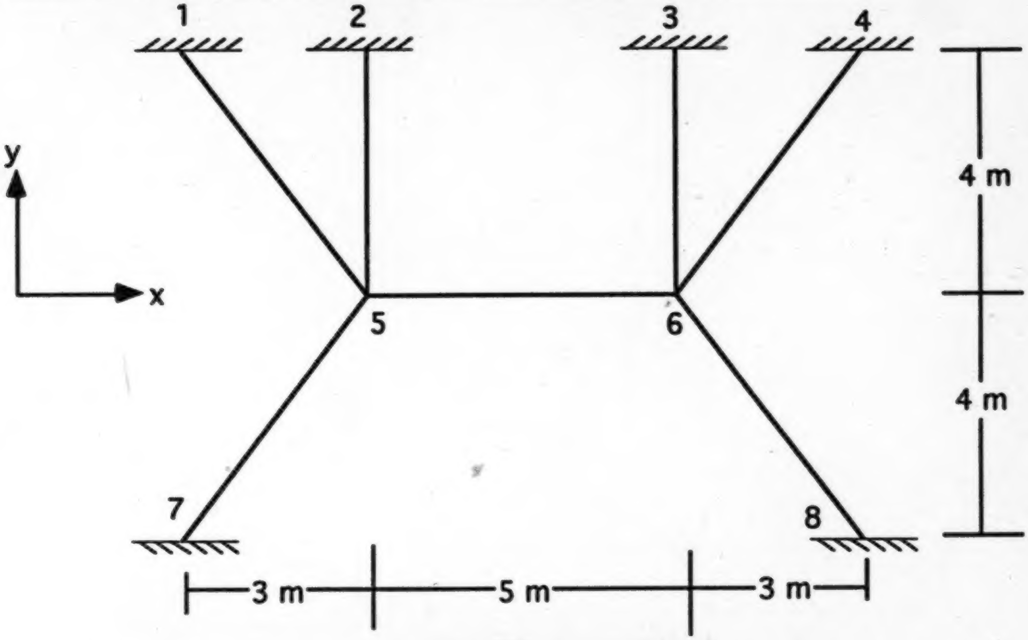
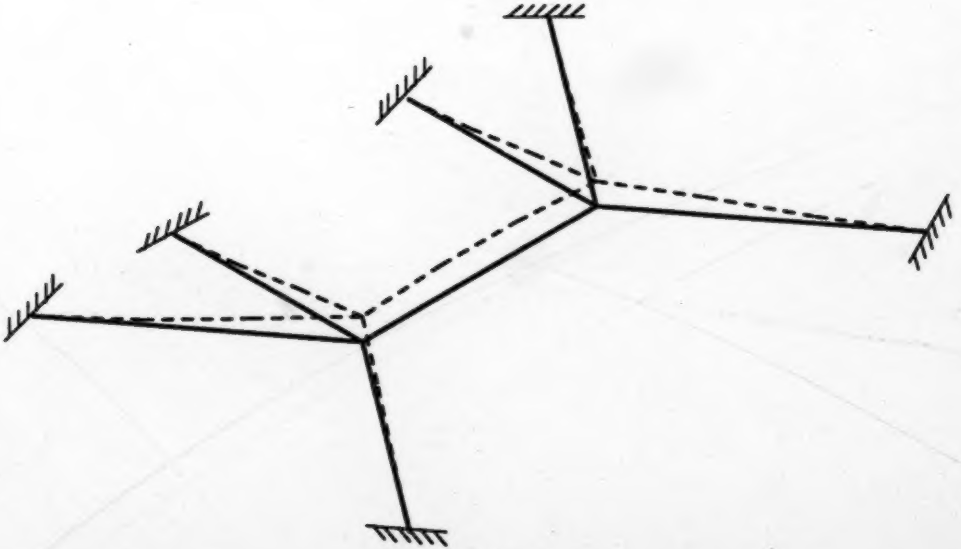


Figure (2-7): Linear out of plane deformation for plane frame with the equal out of plane applied forces at nodes 5, 6.
(a) Geometry of initial configuration



(b) Initial and scaled deformed configurations

Chapter 3

Numerical Method (Dynamic Relaxation)

3.1 Introduction

The Dynamic Relaxation method has a very long history of success in seeking the goal of equilibrium. It has been modified, adapted and enriched in many ways. It has been successfully applied in many material nonlinearity problems to predict the collapse load of a structure. As the DR method needs little effort for the solution, it can easily be combined with a very sophisticated material model to solve highly nonlinear problems. Since 1970 the DR literature has expanded considerably. More complex problems concerned with a large variety of structural configurations have been analyzed using DR. Geometrical and material nonlinearities have been successfully incorporated into the procedure. The technique has been used for nonlinear static analysis of other structures, such as cable networks and membranes. Solutions of plate and shell problems, especially the large deflection case, have also been successfully obtained with DR. As well, a variety of other problems such as the large deformation inelastic response of solids and Elasto-plastic response of nonlinear materials have been solved using DR. For contributions to improvement of the DR method we should mention Underwood [39], which presents a detailed review on the subject of DR, as well as an adaptive DR algorithm for nonlinear problems. Also, the paper by Papadrakakis [22], which presents a method for a automatic evaluation of the DR iteration parameters.

3.2 Theory of Dynamic Relaxation

The discretized equilibrium equations of a structure may be represented in the general form $\mathbf{Ku} = \mathbf{F}$ where $\mathbf{K}$ is the stiffness matrix, $\mathbf{u}$ is the displacement vector and $\mathbf{F}$ is the vector of external forces.

A spatial discretization of the global equations of motion can be represented as a form of Newton's second law as

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{Ku} = \mathbf{F} \tag{3.1}$$

where $\mathbf{M}$ and $\mathbf{C}$ are mass and damping matrices. Henceforth in this chapter dots indicate differentiation with respect to time. The response for this motion is the sum of homogeneous part(transient response) and particular part(steady state response). If the transient part dies out, we are left with particular solution $\mathbf{u}^* = \mathbf{K}^{-1}\mathbf{F}$ which is what we want.

We intend to solve (3.1), for fixed $\mathbf{F}$, in increments of time. For the n^{th} increment this equation can be written as

$$\mathbf{M}\ddot{\mathbf{u}}^{(n)} + \mathbf{C}\dot{\mathbf{u}}^{(n)} + \mathbf{K}\mathbf{u}^{(n)} = \mathbf{F} \tag{3.2}$$

Using the finite difference technique with the central difference scheme we can write

$$\begin{cases} \dot{\mathbf{u}}^{(n-\frac{1}{2})} = \frac{\mathbf{u}^{(n)} - \mathbf{u}^{(n-1)}}{\Delta t} \\ \ddot{\mathbf{u}}^{(n)} = \frac{\dot{\mathbf{u}}^{(n+\frac{1}{2})} - \dot{\mathbf{u}}^{(n-\frac{1}{2})}}{\Delta t} \end{cases} \tag{3.3}$$

where Δt is the fixed time increment. The expression for $\dot{\mathbf{u}}^n$ is obtained by the average

$$\dot{\mathbf{u}}^{(n)} = \frac{\dot{\mathbf{u}}^{(n+\frac{1}{2})} + \dot{\mathbf{u}}^{(n-\frac{1}{2})}}{2} \tag{3.4}$$

Substituting (3.3) and (3.4) into (3.2) we get the following formula

$$\left\{ \begin{array}{l} \dot{\mathbf{u}}^{(n+\frac{1}{2})} = (\frac{1}{\Delta t}\mathbf{M} + \frac{1}{2}\mathbf{C})^{-1}[(\frac{1}{\Delta t}\mathbf{M} - \frac{1}{2}\mathbf{C})\dot{\mathbf{u}}^{(n-\frac{1}{2})} + (\mathbf{F} - \mathbf{K}\mathbf{u}^{(n)})] \\ \mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} + \Delta t \dot{\mathbf{u}}^{(n+\frac{1}{2})} \end{array} \right. \tag{3.5}$$

The initial conditions for DR are of the form [39]

$$\left\{ \begin{array}{l} \mathbf{u}^{(0)} \quad \text{prescribed} \\ \dot{\mathbf{u}}^{(0)} = \mathbf{0} \end{array} \right. \tag{3.6}$$

Using (3.4) and the second of (3.6) gives

$$\dot{\mathbf{u}}^{(-\frac{1}{2})} = -\dot{\mathbf{u}}^{(\frac{1}{2})} \tag{3.7}$$

then from the first of (3.5) we get

$$\dot{\mathbf{u}}^{(\frac{1}{2})} = \frac{\Delta t}{2}\mathbf{M}^{-1}(\mathbf{F} - \mathbf{K}\mathbf{u}^{(0)}) \tag{3.8}$$

Now we can summarize the recurrence formula as

$$\left\{ \begin{array}{ll} \dot{\mathbf{u}}^{(\frac{1}{2})} = \frac{\Delta t}{2}\mathbf{M}^{-1}(\mathbf{F} - \mathbf{K}\mathbf{u}^{(0)}) & \text{for } n = 0 \\ \dot{\mathbf{u}}^{(n+\frac{1}{2})} = (\frac{1}{\Delta t}\mathbf{M} + \frac{1}{2}\mathbf{C})^{-1}[(\frac{1}{\Delta t}\mathbf{M} - \frac{1}{2}\mathbf{C})\dot{\mathbf{u}}^{(n-\frac{1}{2})} + (\mathbf{F} - \mathbf{K}\mathbf{u}^{(n)})] & \text{for } n \neq 0 \\ \mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} + \Delta t \dot{\mathbf{u}}^{(n+\frac{1}{2})} & \text{for all } n \end{array} \right. \tag{3.9}$$

The DR algorithm then can be written as the following steps [39]

1. choose $\mathbf{M}$ and $\mathbf{C}$; $n = 0$; $\mathbf{u}^{(0)}$ given; $\dot{\mathbf{x}}^{(0)} = \mathbf{0}$
2. check residuals $\mathbf{R}^{(n)} = \mathbf{F} - \mathbf{K}\mathbf{u}^{(n)}$ if $\mathbf{R}^{(n)} \approx \mathbf{0}$ stop, otherwise continue
3. if $n = 0$ then $\dot{\mathbf{u}}^{(\frac{1}{2})} = \frac{\Delta t}{2}\mathbf{M}^{-1}\mathbf{R}^{(0)}$ otherwise
   $\dot{\mathbf{u}}^{(n+\frac{1}{2})} = (\frac{1}{\Delta t}\mathbf{M} + \frac{1}{2}\mathbf{C})^{-1}[(\frac{1}{\Delta t}\mathbf{M} - \frac{1}{2}\mathbf{C})\dot{\mathbf{u}}^{(n-\frac{1}{2})} + \mathbf{R}^{(n)}]$
4. $\mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} + \Delta t \dot{\mathbf{u}}^{(n+\frac{1}{2})}$
5. $n = n + 1$; go to 2.

3.3 Application of DR for Linear Deformations

(a) Employing the Finite Difference Method

The equations of motion for a space motion considering linear deformations, ignoring the rotatory inertia ρI, and including the damping terms are

$$\begin{aligned}
\rho A\ddot{u}_1 + C_{u_1}\dot{u}_1 - EAu_1'' &= 0 \\
\rho A\ddot{u}_2 + C_{u_2}\dot{u}_2 + EIu_2'''' &= 0 \\
\rho A\ddot{u}_3 + C_{u_3}\dot{u}_3 + EIu_3'''' &= 0 \\
\rho J\ddot{\theta}_1 + C_{\theta_1}\dot{\theta}_1 - GJ{\theta_1}'' &= 0
\end{aligned} \tag{3.10}$$

where $\theta_1 = a_1$ is the twist angle.

For different classes of nodal constraint given in section (2.3), to get the configuration in each time step, from the previous one, the appropriate boundary conditions at the unconstrained and the constrained nodes should be applied.

(b) Employing the Finite Element Method

Here we give the stiffness matrix **K** for some kind of different joints using FEM (Bathe [4]).

(1) Rigid-Rigid jointed rod in space frame

The column matrices of force and displacement components are

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_i \\ \mathbf{P}_j \end{pmatrix} \quad , \quad \mathbf{d} = \begin{pmatrix} \mathbf{d}_i \\ \mathbf{d}_j \end{pmatrix}$$

where now

$$\mathbf{P}_i = \begin{pmatrix} F_{1i} & F_{2i} & F_{3i} & M_{1i} & M_{2i} & M_{3i} \end{pmatrix} \quad , \quad \mathbf{d}_i = \begin{pmatrix} u_{1i} & u_{2i} & u_{3i} & \theta_{1i} & \theta_{2i} & \theta_{3i} \end{pmatrix}$$

where $\theta_{\alpha i} = a_{\alpha i}$ in which $(\alpha = 1, 2, 3)$. The $\mathbf{P}_j$ and $\mathbf{d}_j$ are defined in similar fashion with suffix j in place of i of course.
Then the stiffness matrix for this case is

$$\mathbf{K} = \begin{pmatrix} k_{11} & Sym. \\ k_{21} & k_{22} \end{pmatrix}$$

where

$$k_{11} = \begin{pmatrix} AE/l & & & & & \\ 0 & 12EI_3/l^3 & & & Sym. & \\ 0 & 0 & 12EI_2/l^3 & & & \\ 0 & 0 & 0 & GJ/l & & \\ 0 & 0 & -6EI_2/l^2 & 0 & 4EI_2/l & \\ 0 & 6EI_3/l^2 & 0 & 0 & 0 & 4EI_3/l \end{pmatrix}$$

$$k_{22} = \begin{pmatrix} AE/l & & & & & \\ 0 & 12EI_3/l^3 & & & Sym. & \\ 0 & 0 & 12EI_2/l^3 & & & \\ 0 & 0 & 0 & GJ/l & & \\ 0 & 0 & 6EI_2/l^2 & 0 & 4EI_2/l & \\ 0 & -6EI_3/l^2 & 0 & 0 & 0 & 4EI_3/l \end{pmatrix}$$

$$k_{21} = \begin{pmatrix} -AE/l & 0 & 0 & 0 & 0 & 0 \\ 0 & -12EI_3/l^3 & 0 & 0 & 0 & -6EI_3/l^2 \\ 0 & 0 & -12EI_2/l^3 & 0 & 6EI_2/l^2 & 0 \\ 0 & 0 & 0 & -GJ/l & 0 & 0 \\ 0 & 0 & -6EI_2/l^2 & 0 & 2EI_2/l & 0 \\ 0 & 6EI_3/l^2 & 0 & 0 & 0 & 2EI_3/l \end{pmatrix}$$

(1) Rigid-Ball jointed rod in space frame

It will be assumed that end i is connected to a rigid joint whereas end j is ball jointed and free of applied moment. The stiffness matrix for this element will be derived by simply modifying the stiffness equations for the rigid-rigid jointed element by using the condition that the moment at end j has zero value. The column matrices of force and displacement components are

$$\mathbf{P}_i = \begin{pmatrix} F_{1i} & F_{2i} & F_{3i} & M_{2i} & M_{3i} \end{pmatrix} \quad , \quad \mathbf{d}_i = \begin{pmatrix} u_{1i} & u_{2i} & u_{3i} & \theta_{2i} & \theta_{3i} \end{pmatrix}$$

and

$$\mathbf{P}_j = \begin{pmatrix} F_{1j} & F_{2j} & F_{3j} \end{pmatrix} \quad , \quad \mathbf{d}_j = \begin{pmatrix} u_{1j} & u_{2j} & u_{3j} \end{pmatrix}$$

Then the stiffness matrix for this case is

$$\mathbf{K} = \begin{pmatrix} AE/l & & & & & & & \\ 0 & 3EI_3/l^3 & & & & & & \\ 0 & 0 & 3EI_2/l^3 & & & Sym. & & \\ 0 & 0 & -3EI_2/l^2 & 3EI_2/l & & & & \\ 0 & 3EI_3/l^2 & 0 & 0 & 3EI_3/l & & & \\ -AE/l & 0 & 0 & 0 & 0 & AE/l & & \\ 0 & -3EI_3/l^3 & 0 & 0 & -3EI_3/l^2 & 0 & 3EI_3/l^3 & \\ 0 & 0 & -3EI_2/l^3 & 3EI_2/l^2 & 0 & 0 & 0 & 3EI_2/l^3 \end{pmatrix}$$

(1) Ball-Ball jointed rod in space frame

The column matrices of force and displacement components simply are

$$\mathbf{P} = \begin{pmatrix} F_{1i} \\ F_{1j} \end{pmatrix} \quad , \quad \mathbf{d} = \begin{pmatrix} u_{1i} \\ u_{1j} \end{pmatrix}$$

Then the stiffness matrix for this case is

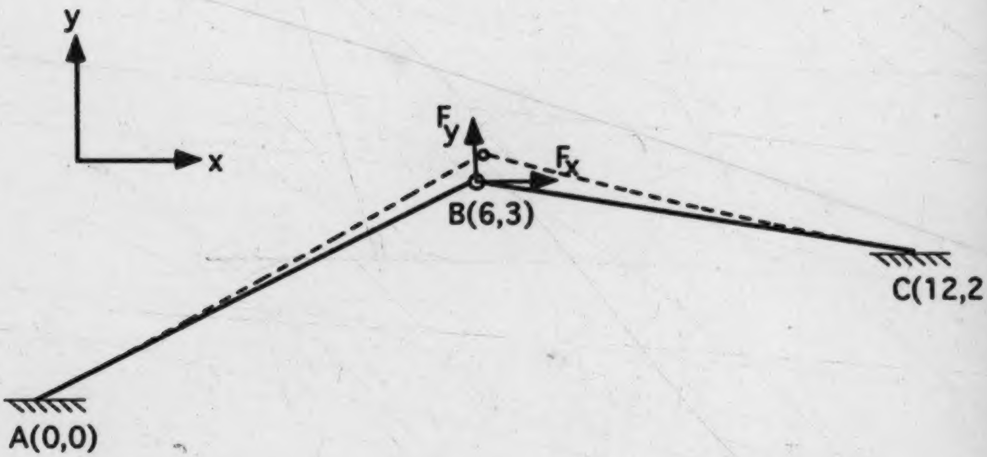
$$\mathbf{K} = \begin{pmatrix} AE/l & -AE/l \\ -AE/l & AE/l \end{pmatrix}$$

Having assembled the stiffness matrix for the lattice and following the given procedure in section (3.2) we could get the deformed configuration for the linear deformation of the lattice.

3.4 Numerical Results

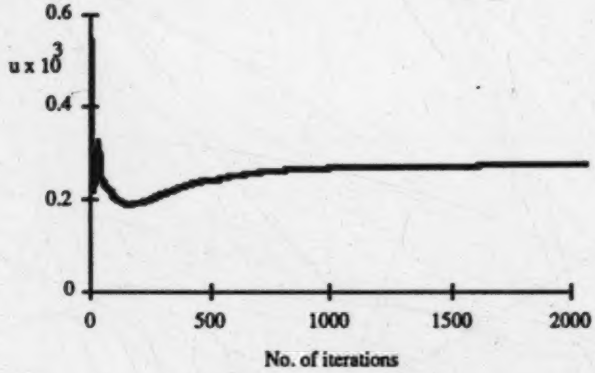
Example (3-1). The linear deformed configuration of the plane frame shown in Figure (3.1a) is derived using FDM with DR. For all members $EA = 1000$, $EI = 1000$. The applied forces are $F_x = 0.1$, $F_y = 0.05$. The graphs (3.1b) and (3.1c) show the time history for horizontal and vertical displacements of node B respectively. As shown in these graphs, after sometime the transient part of the response dies out, and the the steady state of the solution is left which is the solution to the static problem. The scaled deformed configuration is also shown in Figure (3.1a).

Example (3-2). The linear deformed configuration of the space truss shown in Figure (3.2a) is derived using FEM with DR. This truss is also considered in [9] (page 165). For all members $EA = 400\ MN$. The graphs (3.2b) and (3.2c) show the time history for horizontal and vertical displacements of node B respectively. As shown in these graphs, after sometime the transient part of the response dies out, and the the steady state of the solution is left which is the solution to the static problem. We should mention that FEM would reach steady state part much faster comparing to FDM. The scaled deformed configuration is also shown in Figure (3.2a). For displacements node B our results are $u_x = 3.01\ mm$, $u_y = 0.952\ mm$, whereas the results given in [9] are $u_x = 2.962\ mm$, $u_y = 0.94785\ mm$.

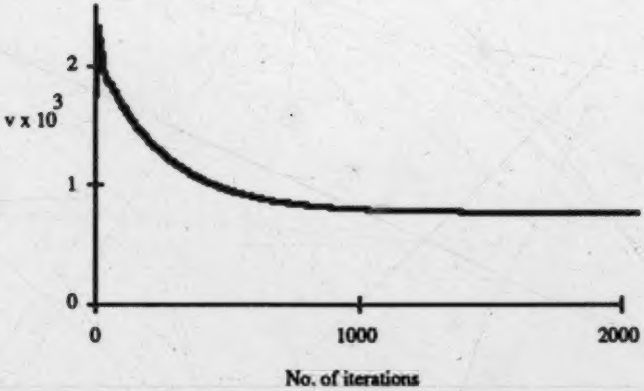


Figure(3-1): Linear deformation of the plane frame with the applied forces at node B using FDM with DR.

(a) Initial configuration and scaled deformed configuration.



(b) Time history for horizontal displacement of node B.



(c) Time history for vertical displacement of node B.

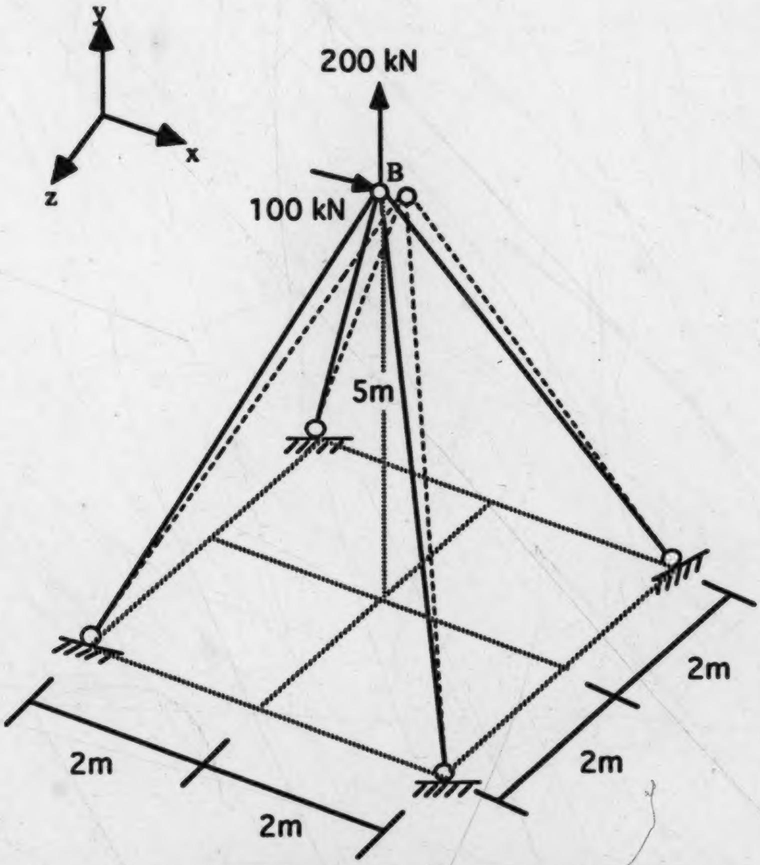
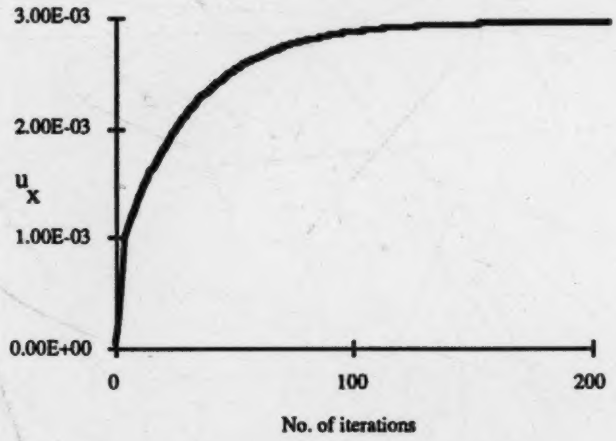
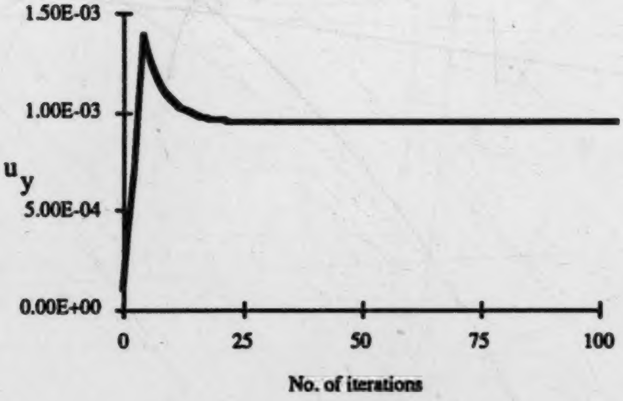

y
x
z
200 kN
B
100 kN
5m
2m
2m
2m
2m

Figure (3-2) : Linear deformation of the space truss with the applied forces at node B using FEM with DR.
(a) Initial configuration and scaled deformed configuration.



(b) Time history for horizontal displacement of node B.



(c) Time history for vertical displacement of node B.

Chapter 4

Large Deformation of Structures by DR

4.1 Dynamics of Rods for Large Deformation in Space

To use DR for nonlinear deformations we derive the equations of motion for large deformation. We only consider rods that are materially homogeneous and composed of isotropic material. For evaluation of the internal terms in the balance laws of linear and angular momentum, we assume that as a rod deforms its cross sections remain planar, undistorted, and normal to the axis. In this dynamical theory, as in the classical theory of the equilibrium of elastic rods, linear constitutive relations implying proportionality of moment and curvature are employed; it is because of the intrinsic nonlinearity of geometrical effects accompanying the flexure and torsion of slender bodies that the resulting field equations are nonlinear.

Now let's find the relation between the orientation of the body-fixed axes ($oe_1e_2e_3$) in the deformed configuration with respect to the body-fixed axes ($OE_1E_2E_3$) in the undeformed configuration using three Eulerian angles. The vectors $\mathbf{e}_2, \mathbf{e}_3$ are along the principal axes of the cross section. Now to find the relation between $\{\mathbf{e}_i\}$ and $\{\mathbf{E}_i\}$, assume the axes ($ox_1y_1z_1$) is rotated about the E_3-axis by an angle θ_3 with respect to the $(OE_1E_2E_3)$. Next $(ox_2y_2z_2)$ is rotated about the y_1-axis by an angle θ_2 with respect to the $(ox_1y_1z_1)$. Finally, $(oe_1e_2e_3)$ is rotated about the x_2-axis by an angle θ_1 with respect to the $(ox_2y_2z_2)$. Hence, we get

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & \sin\theta_1 \\ 0 & -\sin\theta_1 & \cos\theta_1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{pmatrix} \begin{pmatrix} \cos\theta_3 & \sin\theta_3 & 0 \\ -\sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.1}$$

or

$$\mathbf{T} = \begin{pmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{pmatrix} \tag{4.2}$$

where the elements of the transformation matrix $\mathbf{T}$ are

$$T_{11} = \cos\theta_3\cos\theta_2, \qquad T_{12} = \cos\theta_2\sin\theta_3, \qquad T_{13} = -\sin\theta_2$$

$$\begin{aligned} T_{21} &= \cos\theta_3\sin\theta_1\sin\theta_2, & T_{22} &= \sin\theta_1\sin\theta_2\sin\theta_3, & T_{23} &= \cos\theta_2\sin\theta_1 \\ &\quad -\cos\theta_1\sin\theta_3 & &\quad +\cos\theta_1\cos\theta_3 & & \end{aligned} \tag{4.3}$$

$$\begin{aligned} T_{31} &= \cos\theta_1\cos\theta_3\sin\theta_2, & T_{32} &= \cos\theta_1\sin\theta_2\sin\theta_3, & T_{33} &= \cos\theta_1\cos\theta_2 \\ &\quad +\sin\theta_1\sin\theta_3 & &\quad -\cos\theta_3\sin\theta_1 & & \end{aligned}$$

Thus

$$\mathbf{e}_i = T_{ij}\mathbf{E}_j \tag{4.4}$$

Comparing this equation to Eqn.(2.11) we see that

$$\mathbf{T} = \mathbf{R}^T \tag{4.5}$$

To describe the motion of the rod, let $\mathbf{r}(S,t)$ be the position vector of a point on the deformed centroidal line and $\mathbf{r}_p(S,Y,Z,t)$ be the position vector of any other point p on the corresponding deformed cross section at time t. Where Y, Z are the coordinates of the point p in the principal axes directions (Bernard *et al.*[7]). We can write

$$\mathbf{r}_p(S,Y,Z,t) = \mathbf{r}(S,t) + Y\mathbf{e}_2(S,t) + Z\mathbf{e}_3(S,t) \tag{4.6}$$

Let the Piola stress vector at the point p be $\boldsymbol{\sigma} = \boldsymbol{\sigma}(S,Y,Z,t)$. The *resultant force* at $X, \mathbf{F}(S,t)$, is the integral of the vector field $\boldsymbol{\sigma}$ over the area

$$\mathbf{F}(S,t) = \iint\limits_{area} \boldsymbol{\sigma} dY dZ \tag{4.7}$$

The *resultant moment* of $\boldsymbol{\sigma}$ about the centeriod point is

$$\mathbf{M}(S,t) = \iint\limits_{area} (Y\mathbf{e}_2 + Z\mathbf{e}_3) \times \boldsymbol{\sigma} dY dZ \tag{4.8}$$

If body forces and tractions on the lateral surfaces are negligible, the balance laws for linear and angular momentum including the damping terms yield

$$\mathbf{F}' = \iint\limits_{area} \rho \ddot{\mathbf{r}}_p dY dZ + \iint\limits_{area} c\dot{\mathbf{r}}_p dY dZ \tag{4.9}$$

and

$$\mathbf{M}' + \mathbf{r}' \times \mathbf{F} = \iint\limits_{area} \rho(Y\mathbf{e}_2 + Z\mathbf{e}_3) \times \ddot{\mathbf{r}}_p dY dZ + \iint\limits_{area} c(Y\mathbf{e}_2 + Z\mathbf{e}_3) \times \dot{\mathbf{r}}_p dY dZ \tag{4.10}$$

Here ρ is the mass density and c is the damping coefficient per volume in the reference configuration and both are taken to be constant in space and time. Substituting for $\mathbf{r}_p$ and noting that Y and Z are not functions of time, Eqns.(4.9) and (4.10) reduce to

$$\mathbf{F}' = \rho A\ddot{\mathbf{r}} + cA\dot{\mathbf{r}} \tag{4.11}$$

and

$$\mathbf{M}' + \mathbf{r}' \times \mathbf{F} = \rho I_3 \mathbf{e}_2 \times \ddot{\mathbf{e}}_2 + \rho I_2 \mathbf{e}_3 \times \ddot{\mathbf{e}}_3 + c I_3 \mathbf{e}_2 \times \dot{\mathbf{e}}_2 + c I_2 \mathbf{e}_3 \times \dot{\mathbf{e}}_3 \tag{4.12}$$

where

$$I_2 = \iint_{area} Z^2 dY dZ \quad , \quad I_3 = \iint_{area} Y^2 dY dZ \tag{4.13}$$

The rods considered here will be assumed to be linearly elastic and have a quadratic strain energy function, w, per unit length of the reference placement. It is usually further assumed that w depends only on the difference in curvature components of the rod (i.e. κ_2 and κ_3 as well as the twist per unit length κ_1) between the deformed and undeformed (reference) configurations (Love [19]). In the rod's undeformed configuration κ_i take on the values κ_i^0 and we have

$$w(\lambda, \boldsymbol{\kappa}) = \frac{1}{2}EI_2(\kappa_2 - \kappa_2^0)^2 + \frac{1}{2}EI_3(\kappa_3 - \kappa_3^0)^2 + \frac{1}{2}GJ(\kappa_1 - \kappa_1^0)^2 + \frac{1}{2}EA(\lambda - 1)^2 \tag{4.14}$$

where

$$J = \iint_{area} (Y^2 + Z^2 + Y\frac{\partial \phi}{\partial Z} - Z\frac{\partial \phi}{\partial Y}) dY dZ \tag{4.15}$$

with ϕ the warping function for torsion given by the linear theory of elasticity. For rods of circular cross section, ϕ vanishes.

Using Eqns.(2.27), (2.26) and (4.14) yields

$$\mathbf{M} = EI_2(\kappa_2 - \kappa_2^0)\mathbf{e}_2 + EI_3(\kappa_3 - \kappa_3^0)\mathbf{e}_3 + GJ(\kappa_1 - \kappa_3^0)\mathbf{e}_1 \tag{4.16}$$

Also using Eqn.(2.54) and Eqn.(2.76) results in

$$\mathbf{F} = F_i \mathbf{e}_i \qquad \text{where} \qquad F_1 = EA(\lambda - 1) \tag{4.17}$$

Also we have

$$\mathbf{r} = r_i \mathbf{E}_i \qquad \text{and} \qquad \mathbf{r}' = r_i' \mathbf{E}_i \tag{4.18}$$

Using Eqn.(2.2) and (4.4) gives

$$r'_i = \lambda T_{1i} \qquad \text{where} \qquad \lambda = |\mathbf{r}'| \tag{4.19}$$

Using Eqn.(2.15) and since $k_i = w_i$ we get

$$\begin{aligned} \mathbf{e'}_1 &= \kappa_3 \mathbf{e}_2 - \kappa_2 \mathbf{e}_3 \\ \mathbf{e'}_2 &= \kappa_1 \mathbf{e}_3 - \kappa_3 \mathbf{e}_1 \\ \mathbf{e'}_3 &= \kappa_2 \mathbf{e}_1 - \kappa_1 \mathbf{e}_2 \end{aligned} \tag{4.20}$$

Taking the derivative of Eqns.(4.16), (4.17) and also using Eqns. (4.20) yields

$$\begin{aligned} \mathbf{M}' =& GJ\kappa'_1\mathbf{e}_1 + EI_2\kappa'_2\mathbf{e}_2 + EI_3\kappa'_3\mathbf{e}_3 + GJ(\kappa_1 - \kappa_1^0)(\kappa_3\mathbf{e}_2 - \kappa_2\mathbf{e}_3) \\ &+ EI_2(\kappa_2 - \kappa_2^0)(\kappa_1\mathbf{e}_3 - \kappa_3\mathbf{e}_1) + EI_3(\kappa_3 - \kappa_3^0)(\kappa_2\mathbf{e}_1 - \kappa_1\mathbf{e}_2) \end{aligned} \tag{4.21}$$

and

$$\begin{aligned} \mathbf{F}' =& F'_i\mathbf{e}_i + F_1(\kappa_3\mathbf{e}_2 - \kappa_2\mathbf{e}_3) + F_2(\kappa_1\mathbf{e}_3 - \kappa_3\mathbf{e}_1) \\ &+ F_3(\kappa_2\mathbf{e}_1 - \kappa_1\mathbf{e}_2) \end{aligned} \tag{4.22}$$

To make the calculations easier we express the Eqn.(4.22) in the reference coordinate $\{\mathbf{E}_i\}$ and Eqn.(4.21) in the body-fixed axes in the deformed configuration $\{\mathbf{e}_i\}$. To express Eqn.(4.22) in the reference coordinate $\{\mathbf{E}_i\}$ using (4.4) gives

$$\mathbf{F}' = T_{ij}\eta_i\mathbf{E}_j \tag{4.23}$$

where

$$\begin{aligned} \eta_1 &= F'_1 - F_2\kappa_3 + F_3\kappa_2 \\ \eta_2 &= F'_2 - F_3\kappa_1 + F_1\kappa_3 \\ \eta_3 &= F'_3 - F_1\kappa_2 + F_2\kappa_1 \end{aligned} \tag{4.24}$$

Combining Eqns. (4.11) and (4.23) results in

$$T_{ij}\eta_i\mathbf{E}_j = \rho A\ddot{r}_j\mathbf{E}_j + cA\dot{r}_j\mathbf{E}_j \qquad \text{where} \qquad j = 1, 2, 3 \tag{4.25}$$

Taking the dot product of the Eqn.(4.12) with $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ respectively, gives

$$\begin{aligned}
\mathbf{M}'.\mathbf{e}_1 &= \rho I_3\mathbf{e}_3.\ddot{\mathbf{e}}_2 - \rho I_2\mathbf{e}_2.\ddot{\mathbf{e}}_3 + cI_3\mathbf{e}_3.\dot{\mathbf{e}}_2 - cI_2\mathbf{e}_2.\dot{\mathbf{e}}_3 \\
\mathbf{M}'.\mathbf{e}_2 - \lambda F_3 &= \rho I_2\mathbf{e}_1.\ddot{\mathbf{e}}_3 + cI_2\mathbf{e}_1.\dot{\mathbf{e}}_3 \\
\mathbf{M}'.\mathbf{e}_3 + \lambda F_2 &= -\rho I_3\mathbf{e}_1.\ddot{\mathbf{e}}_2 - cI_3\mathbf{e}_1.\dot{\mathbf{e}}_2
\end{aligned} \tag{4.26}$$

Substituting Eqn.(4.21) into the equations of motion (4.26) yields

$$\begin{aligned}
GJ\kappa_1' + (EI_3 - EI_2)\kappa_2\kappa_3 + EI_2\kappa_3\kappa_2^0 - EI_3\kappa_2\kappa_3^0 &= \rho I_3\mathbf{e}_3.\ddot{\mathbf{e}}_2 - \rho I_2\mathbf{e}_2.\ddot{\mathbf{e}}_3 \\
&\quad + cI_3\mathbf{e}_3.\dot{\mathbf{e}}_2 - cI_2\mathbf{e}_2.\dot{\mathbf{e}}_3 \\
EI_2\kappa_2' + (GJ - EI_3)\kappa_1\kappa_3 - GJ\kappa_3\kappa_1^0 + EI_3\kappa_1\kappa_3^0 - \lambda F_3 &= \rho I_2\mathbf{e}_1.\ddot{\mathbf{e}}_3 + cI_2\mathbf{e}_1.\dot{\mathbf{e}}_3 \\
EI_3\kappa_3' + (EI_2 - GJ)\kappa_1\kappa_2 + GJ\kappa_2\kappa_1^0 - EI_2\kappa_1\kappa_2^0 + \lambda F_2 &= -\rho I_3\mathbf{e}_1.\ddot{\mathbf{e}}_2 - cI_3\mathbf{e}_1.\dot{\mathbf{e}}_2
\end{aligned} \tag{4.27}$$

Using Eqns.(2.18) and (2.13) results in

$$\begin{aligned}
\kappa_1 &= \frac{1}{2}(\mathbf{e}'_2 \cdot \mathbf{e}_3 - \mathbf{e}_2 \cdot \mathbf{e}'_3) \\
\kappa_2 &= \frac{1}{2}(\mathbf{e}'_3 \cdot \mathbf{e}_1 - \mathbf{e}_3 \cdot \mathbf{e}'_1) \\
\kappa_3 &= \frac{1}{2}(\mathbf{e}'_1 \cdot \mathbf{e}_2 - \mathbf{e}_1 \cdot \mathbf{e}'_2)
\end{aligned} \tag{4.28}$$

Also substituting (4.4) into (4.28) yields

$$\begin{aligned}
\kappa_1 &= \frac{1}{2}(T'_{2i}T_{3i} - T_{2j}T'_{3j}) \\
\kappa_2 &= \frac{1}{2}(T'_{3i}T_{1i} - T_{3j}T'_{1j}) \\
\kappa_3 &= \frac{1}{2}(T'_{1i}T_{2i} - T_{1j}T'_{2j})
\end{aligned} \tag{4.29}$$

Taking derivative of the elements of the transformations matrix $\mathbf{T}$, Eqns.(4.3), and substituting them into Eqns. (4.29) and simplifying them results in

$$\begin{aligned}
\kappa_1 &= \theta_1' - \theta_3' \sin\theta_2 \\
\kappa_2 &= \theta_2' \cos\theta_1 + \theta_3' \sin\theta_1 \cos\theta_2 \\
\kappa_3 &= -\theta_2' \sin\theta_1 + \theta_3' \cos\theta_1 \cos\theta_2
\end{aligned} \tag{4.30}$$

Taking derivative of these Eqns. also gives

$$\begin{aligned}
\kappa_1' &= \theta_1'' - \theta_3'' \sin\theta_2 - \theta_3'\theta_2' \cos\theta_2 \\
\kappa_2' &= \theta_2'' \cos\theta_1 - \theta_2'\theta_1' \sin\theta_1 + \theta_3'' \sin\theta_1 \cos\theta_2 \\
&\quad + \theta_3'\theta_1' \cos\theta_1 \cos\theta_2 - \theta_3'\theta_2' \sin\theta_1 \sin\theta_2 \qquad (4.31) \\
\kappa_3' &= -\theta_2'' \sin\theta_1 - \theta_2'\theta_1' \cos\theta_1 + \theta_3'' \cos\theta_1 \cos\theta_2 \\
&\quad - \theta_3'\theta_1' \sin\theta_1 \cos\theta_2 - \theta_3'\theta_2' \cos\theta_1 \sin\theta_2
\end{aligned}$$

4.2 Two-Dimensional Large Deformation (Using DR Method)

We consider the two-dimensional large deformation of a structure consisting of initially straight rods ($\kappa^0 = 0$) with circular cross-sections ($EI_2 = EI_3 = EI$) in which the axis of each rod lies at all time in the ($\mathbf{E}_1, \mathbf{E}_2$) plane. In such a deformation κ_3 is non-zero, while κ_1 and κ_2 are zero. Using (4.3) gives

$$\begin{aligned}
T_{11} &= \cos\theta_3 & T_{12} &= \sin\theta_3 \\
T_{21} &= -\sin\theta_3 & T_{22} &= \cos\theta_3
\end{aligned} \qquad (4.32)$$

Using Eqns.(4.29) with (4.32) results in

$$\kappa_1 = \kappa_2 = 0 \qquad \kappa_3 = \theta_3' \qquad (4.33)$$

Taking $\theta = \theta_3$ and substituting Eqns.(4.33) into (4.20) gives

$$\mathbf{e'}_1 = \theta' \mathbf{e}_2, \qquad \mathbf{e'}_2 = -\theta' \mathbf{e}_1 \qquad (4.34)$$

Using (4.19) with (4.32) gives

$$r_1' = \lambda \cos\theta, \qquad r_2' = \lambda \sin\theta \qquad \text{where} \qquad \lambda = |\mathbf{r}'| \qquad (4.35)$$

Substitution Eqns.(4.33) into Eqns.(4.25) and (4.27) and simplifying yields

$$EI_3\theta'' + \lambda F_2 = \rho I \ddot{\theta} + cI\dot{\theta} \qquad (4.36)$$

and

$$\begin{aligned} F_1'\cos\theta - \theta' F_1 \sin\theta - F_2'\sin\theta - \theta' F_2\cos\theta &= \rho A\ddot{r}_1 + cA\dot{r}_1 \\ F_1'\sin\theta + \theta' F_1 \cos\theta + F_2'\cos\theta - \theta' F_2\sin\theta &= \rho A\ddot{r}_2 + cA\dot{r}_2 \end{aligned} \tag{4.37}$$

For the thin rods we consider the form of the equilibrium equation of (4.36) which becomes

$$EI\theta'' + \lambda F_2 = 0 \tag{4.38}$$

We intend to solve (4.37) in increments of time. At the n^{th} increment this equation can be written as

$$\begin{aligned} m_{r_1}\ddot{r}_1^{(n)} + c_{r_1}\dot{r}_1^{(n)} - (F_1'\cos\theta - \theta' F_1 \sin\theta - F_2'\sin\theta - \theta' F_2\cos\theta)^{(n)} = 0 \\ m_{r_2}\ddot{r}_1^{(n)} + c_{r_2}\dot{r}_2^{(n)} - (F_1'\sin\theta + \theta' F_1 \cos\theta + F_2'\cos\theta - \theta' F_2\sin\theta)^{(n)} = 0 \end{aligned} \tag{4.39}$$

where m_{r_1}, m_{r_2} and c_{r_1}, c_{r_2} are virtual mass and damping coefficients. In DR we are not considering the real dynamics of the problem and the goal is achieving the solution of the equilibrium equations. So the mass and damping coefficients are not necessarily those of the structure and they should be chosen in such a way that the number of iterations required for convergence is minimized.

The DR algorithm then can be used as the following steps

1. choose $n = 0; r_1$ and r_2 given; $\dot{r}_1 = 0$ and $\dot{r}_2 = 0$
2. use $\theta = \arctan\frac{r_2'}{r_1'}$ and $\lambda = |\mathbf{r}'|$ to get θ and λ at the n^{th} step
3. use $F_2 = -\frac{EI\theta''}{\lambda}$ and $F_1 = EA(\lambda - 1)$ to get the shear and axial forces at the n^{th} step
4. check R_1 and R_2 at the n^{th} step
   $R_1^{(n)} = (F_1'\cos\theta - \theta' F_1 \sin\theta - F_2'\sin\theta - \theta' F_2\cos\theta)^{(n)}$
   $R_2^{(n)} = (F_1'\sin\theta + \theta' F_1 \cos\theta + F_2'\cos\theta - \theta' F_2\sin\theta)^{(n)}$
   if $R_1^{(n)} \approx 0$ and $R_2^{(n)} \approx 0$ stop, otherwise continue

5. if $n = 0$ then $\dot{r}_1^{(\frac{1}{2})} = \frac{\Delta t}{2m_{r_1}} R_1^{(0)}$ and $\dot{r}_2^{(\frac{1}{2})} = \frac{\Delta t}{2m_{r_2}} R_2^{(0)}$ otherwise
   $\dot{r}_1^{(n+\frac{1}{2})} = (\frac{1}{\Delta t} m_{r_1} + \frac{1}{2} c_{r_1})^{-1}[(\frac{1}{\Delta t} m_{r_1} - \frac{1}{2} c_{r_1})\dot{r}_1^{(n-\frac{1}{2})} + R_1^{(n)}]$
   $\dot{r}_2^{(n+\frac{1}{2})} = (\frac{1}{\Delta t} m_{r_2} + \frac{1}{2} c_{r_2})^{-1}[(\frac{1}{\Delta t} m_{r_2} - \frac{1}{2} c_{r_2})\dot{r}_2^{(n-\frac{1}{2})} + R_2^{(n)}]$
6. ${r_1}^{(n+1)} = {r_1}^{(n)} + \Delta t \dot{r}_1^{(n+\frac{1}{2})}$ and ${r_2}^{(n+1)} = {r_2}^{(n)} + \Delta t \dot{r}_2^{(n+\frac{1}{2})}$
7. $n = n + 1$; go to 2.

4.3 Three-Dimensional Large Deformation (Using DR Method)

Consider a structure consisting of rods which are initially curved and twisted. For rods with the circular cross-section ($EI_2 = EI_3 = EI$), simplifying equations of motion (4.27) gives

$$\begin{aligned} &GJ\kappa_1' + EI(\kappa_3\kappa_2^0 - \kappa_2\kappa_3^0) = \rho I(\mathbf{e}_3.\ddot{\mathbf{e}}_2 - \mathbf{e}_2.\ddot{\mathbf{e}}_3) + cI(\mathbf{e}_3.\dot{\mathbf{e}}_2 - \mathbf{e}_2.\dot{\mathbf{e}}_3) \qquad (4.40) \\ &EI\kappa_2' + (GJ - EI)\kappa_1\kappa_3 - GJ\kappa_3\kappa_1^0 + EI\kappa_1\kappa_3^0 - \lambda F_3 = \rho I\mathbf{e}_1.\ddot{\mathbf{e}}_3 + cI\mathbf{e}_1.\dot{\mathbf{e}}_3 \\ &EI\kappa_3' + (EI - GJ)\kappa_1\kappa_2 + GJ\kappa_2\kappa_1^0 - EI\kappa_1\kappa_2^0 + \lambda F_2 = -\rho I\mathbf{e}_1.\ddot{\mathbf{e}}_2 - cI\mathbf{e}_1.\dot{\mathbf{e}}_2 \end{aligned}$$

Taking derivative of Eqn.(4.4) with respect to time and substituting them into the first equation of (4.27) and simplifying yields

$$\begin{aligned} GJ\kappa_1' + EI(\kappa_3\kappa_2^0 - \kappa_2\kappa_3^0) =& \rho I(-2\dot{\theta}_2\dot{\theta}_3 cos\theta_2 - 2\ddot{\theta}_3 sin\theta_2 + 2\ddot{\theta}_1) \\ &+ cI(2\dot{\theta}_1 - 2\dot{\theta}_3 sin\theta_2) \qquad (4.41) \end{aligned}$$

Rearranging this yields

$$m_{\theta_1}\ddot{\theta}_1 + c_{\theta_1}\ddot{\theta}_1 - (\frac{GJ}{2}\kappa_1' + \frac{EI}{2}(\kappa_3\kappa_2^0 - \kappa_2\kappa_3^0) + m_{\theta_1}(\dot{\theta}_2\dot{\theta}_3 cos\theta_2 + \ddot{\theta}_3 sin\theta_2) + c_{\theta_1}\dot{\theta}_3 sin\theta_2) = 0 \qquad (4.42)$$

We intend to solve (4.25) and (4.42) in increments of time. At the n^{th} increment Eqn. (4.25) can be written as

$$m_{r_k}\ddot{r}_k^{(n)} + c_{r_k}\dot{r}_k^{(n)} - (T_{ik}\eta_i)^{(n)} = 0\ ; \quad \text{where } k = 1, 2, 3 \qquad (4.43)$$

where m_{r_1}, m_{r_2}, m_{r_3}, m_{θ_1} and c_{r_1}, c_{r_2}, c_{r_3}, c_{θ_1} are virtual mass and damping coefficients and should be chosen in such a way that the number of iterations required for convergence is minimized.

The DR algorithm then can be used as the following steps

1. Choose $n = 0; r_1$, r_2, r_3 and θ_1 given; $\dot{r}_1 = 0$, $\dot{r}_2 = 0$, $\dot{r}_3 = 0$ and $\dot{\theta}_1 = 0$
2. Use $\lambda = |\mathbf{r}'|$, $\theta_2 = \arcsin \frac{-r'_3}{\lambda}$ and $\theta_3 = \arctan \frac{r'_2}{r'_1}$ to get λ, θ_2 and θ_3 at the n^{th} step
3. Having θ_1, θ_2 and θ_3 , using Eqns. (4.30) and (4.31) calculate κ_1, κ_2, κ_3 and κ'_1, κ'_2, κ'_3 at the n^{th} step
4. Using backward finite difference equations in time, calculate $\dot{\theta}_2$, $\dot{\theta}_3$ and $\ddot{\theta}_2$, $\ddot{\theta}_3$.
5. Calculate the residual $R_{\theta_1}^{(n)}$ at the n^{th} step from

   $$R_{\theta_1}^{(n)} = (\frac{GJ}{2}\kappa'_1 + \frac{EI}{2}(\kappa_3\kappa_2^0 - \kappa_2\kappa_3^0) + m_{\theta_1}(\dot{\theta}_2\dot{\theta}_3 cos\theta_2 + \ddot{\theta}_3 sin\theta_2) + c_{\theta_1}\dot{\theta}_3 sin\theta_2)$$

6. Calculate

   $$\begin{cases} \dot{\theta}_1^{(\frac{1}{2})} = \frac{\Delta t}{2m_{\theta_1}} R_{\theta_1}^{(0)} & \text{for } n = 0 \\ \dot{\theta}_1^{(n+\frac{1}{2})} = (\frac{1}{\Delta t} m_{\theta_1} + \frac{1}{2} c_{\theta_1})^{-1}[(\frac{1}{\Delta t} m_{\theta_1} - \frac{1}{2} c_{\theta_1})\dot{\theta}_1^{(n-\frac{1}{2})} + R_{\theta_1}^{(n)}] & \text{for } n \neq 0 \\ {\theta_1}^{(n+1)} = {\theta_1}^{(n)} + \Delta t \dot{\theta}_1^{(n+\frac{1}{2})} & \text{for all } n \end{cases}$$

7. Check residuals R_{r_j} at n^{th} step for $j = 1, 2, 3$ from
   $R_{r_j}^{(n)} = (T_{ij}\eta_i)^{(n)}$
   if $R_{r_1}^{(n)} \approx 0$, $R_{r_2}^{(n)} \approx 0$, $R_{r_3}^{(n)} \approx 0$ and $R_{\theta_1}^{(n)} \approx 0$ stop, otherwise continue
8. For $j = 1, 2, 3$; calculate

   $$\begin{cases} \dot{r}_j^{(\frac{1}{2})} = \frac{\Delta t}{2m_{r_j}} R_{r_j}^{(0)} & \text{for } n = 0 \\ \dot{r}_j^{(n+\frac{1}{2})} = (\frac{1}{\Delta t} m_{r_j} + \frac{1}{2} c_{r_j})^{-1}[(\frac{1}{\Delta t} m_{r_j} - \frac{1}{2} c_{r_j})\dot{r}_j^{(n-\frac{1}{2})} + R_{r_j}^{(n)}] & \text{for } n \neq 0 \\ {r_j}^{(n+1)} = {r_j}^{(n)} + \Delta t \dot{r}_j^{(n+\frac{1}{2})} & \text{for all } n \end{cases}$$

9. $n = n + 1$; go to 2.

4.4 Numerical Results

Example (4-1): The nonlinear deformation for a simple cantilever beam is compared in Figures (4.1a) and (4.1b) considering Newton-Raphson and Dynamic Relaxation respectively. For the simplicity, for the property of beam we considered $EA = EI = 1$ also the length of beam is assumed 1. The applied forces are $F_{X_1} = F_{X_2} = 1$. In undeformed configuration the components of position vector at free end, are considered $X_1 = 1$, $X_2 = 0$. In deformed configuration the components of position vector at free end, with DR becomes $r_1 = 2.09133$ and $r_2 = 0.8232$, whereas with NR becomes $r_1 = 2.125$ and $r_2 = 0.8195$. Using DR the time history of horizontal and vertical position for the free end is shown in Figures (4.1c) and (4.1d). As shown in these graphs, after sometime the transient part of the response dies out, and the the steady state of the solution is left which is the solution to the static problem. Also the time history of λ for free end and distribution of λ for final result as a function of initial arclength in a reference placement (S) is given in Figures (4.1e) and (4.1f).

Example (4-2): The two member truss shown in Figure (4.2a) was used by many investigators to test the mathematical formulation and the numerical solution procedure. Hangai and Kawamata [15] have previously analyzed this truss by the static perturbation technique using various degrees of approximation in force-displacement relation. Papadrakakis [23] solved this problem by DR method using a *beam-column* approach in the element formulation. By assuming that each member is going to be straight in the deformation, there would be an analytical solution for this truss (Bathe [4]). The results obtained with the present approach appeared to be in complete agreement previous results. The deformed configuration after the snap through is shown in Figure (4.2b). In the load control for the increasing load Figure (4.2c) and decreasing load Figure (4.2d) DR does not pick up the unstable branch, so there is a jump in the graphs. In the displacement control with increasing the prescribed displacement for node a DR gives the the whole load-displacement graph Figure (4.2e) as the stable solution. In displacement control we specify the displacement for a specific node (here node a), usually step by step, then using DR we derive the displacements for the rest of the structure for that given specified displacement. The external required force for equilibrium would be the negative of resultant of the internal forces at the specified node (here node a).

Example (4-3): William's toggle frame. This problem, shown in Figure (4.3a) has been solved analytically and tested experimentally by Williams [41]. In his analytical treatment of the frame, Williams took into consideration the finite change of geometry as well as the effects of the axial forces on the flexural stiffness and the flexural shortening of the members. Papadrakakis [23] used the beam-column approach to derive the nonlinear equilibrium equations, which were solved by DR procedure. Wood and Zienkiewicz [42] have also investigated this problem employing an assumed displacement finite element approach with 5 element per member. Meek and Tan [20] also used a beam-column large rotation formation and Crisfield's constant-arc length method. The deformed configuration after the snap through is shown in Figure (4.3b). In the load control for the increasing load Figure (4.3c) and decreasing load Figure (4.3d) DR does not pick up the unstable branch, so there is a jump in the graphs. In the displacement control with increasing the prescribed displacement for node a DR gives the the whole load-displacement graph Figure (4.3e) as the stable solution. In displacement control we specify the displacement for a specific node (here node a), usually step by step, then using DR we derive the displacements for the rest of the structure for that given specified displacement. The external required force for equilibrium would be the negative of resultant of the internal forces at the specified node (here node a).

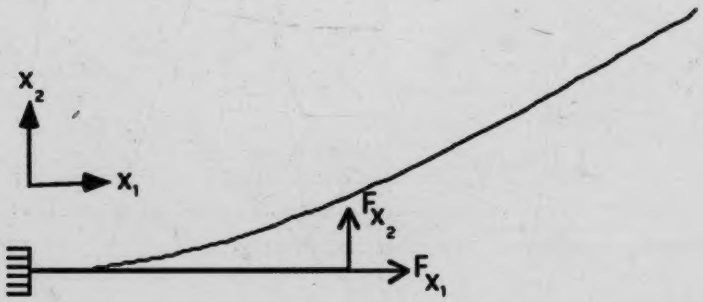
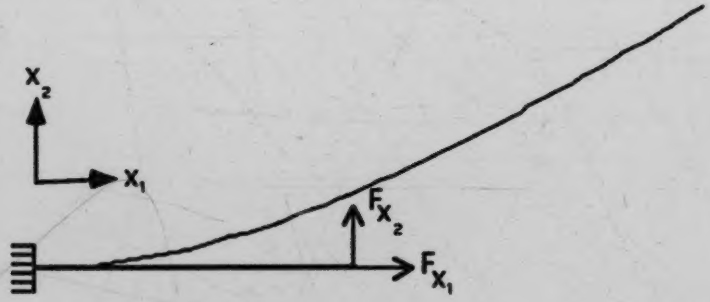
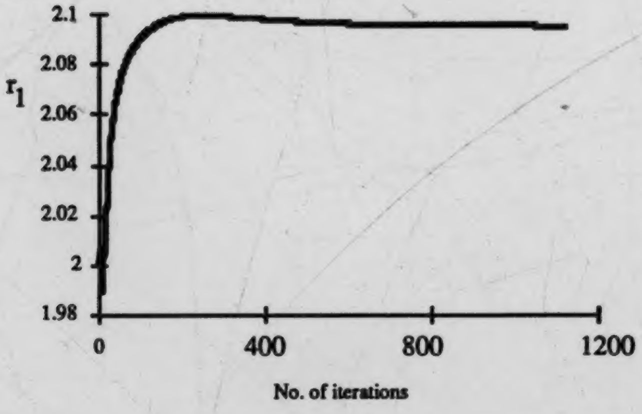


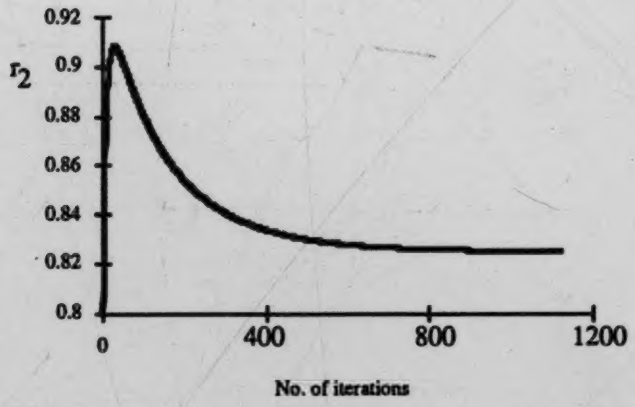
Figure (4-1): Non-linear deformation of a cantilever beam with applied forces at free end .
(a) The deformed configuration is derived using FDM with Newton-Raphson method.



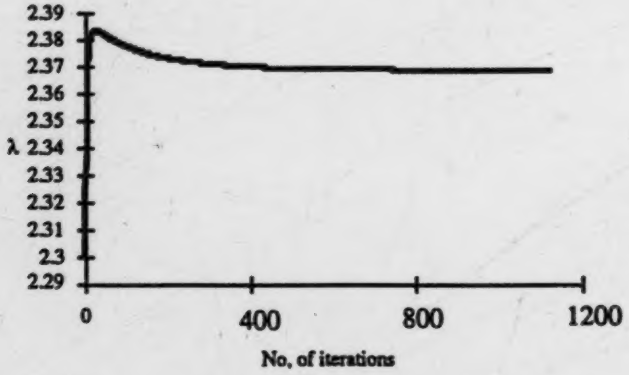
(b) The deformed configuration is derived using FDM with DR.



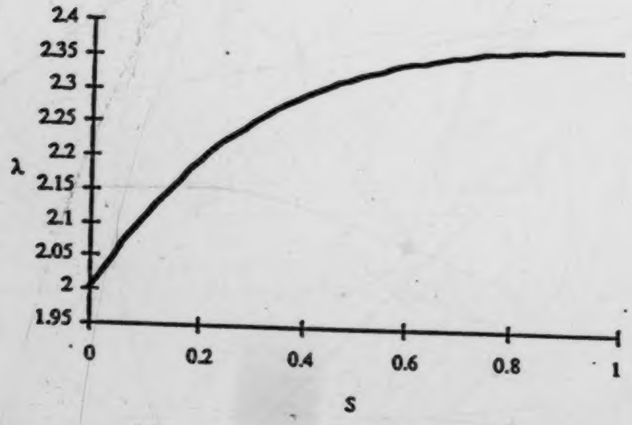
(c) Time history of r_1 (horizontal position) for the free end.



(d) Time history of r_2 (vertical position) for the free end.



(e) Time history of λ for the free end.



(f) Distribution of λ along the beam in the deformed configuration.

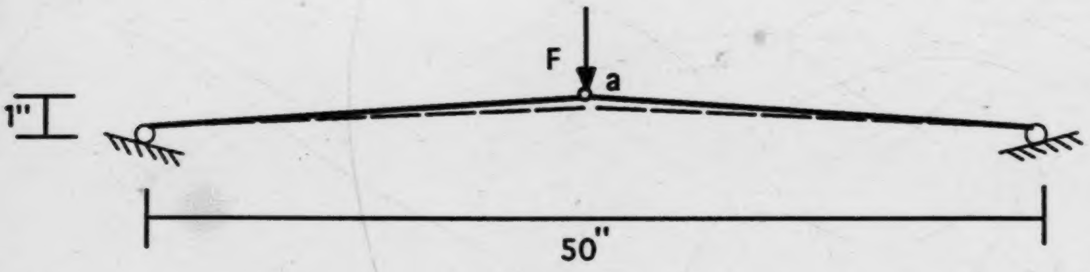
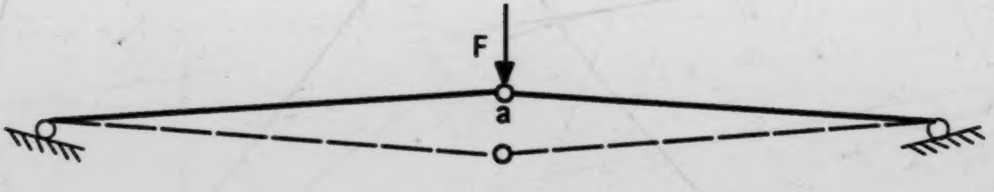


Figure (4-2): The two-member truss with the vertical force applied at pin-jointed node a.
(a) The initial and deformed configurations (before the snap through).



(b) The initial and deformed configurations (after the snap through).

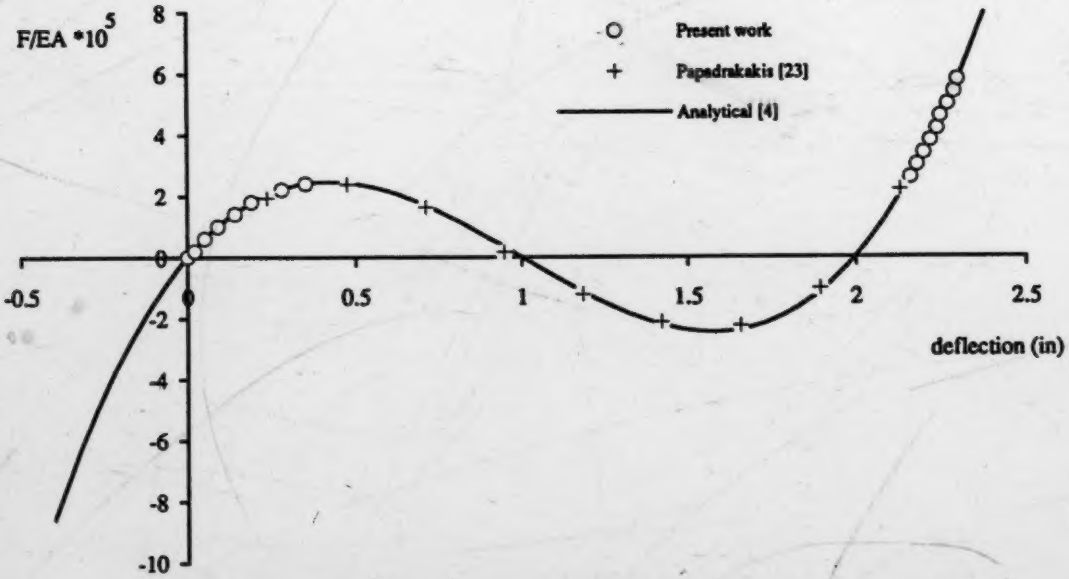


Figure (4-2c) Analytical and numerical results of load-deflection curve for node a . (increasing load)

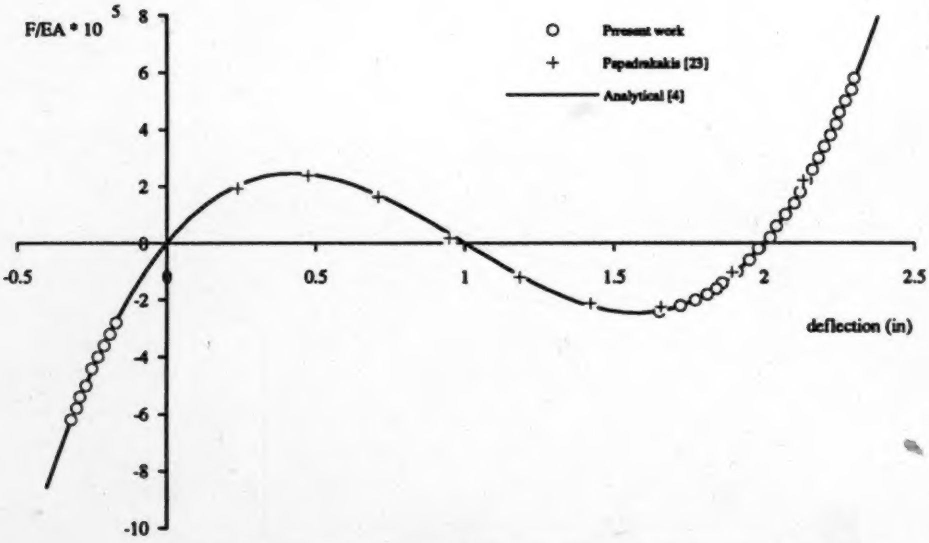


Figure (4-2d) Analytical and numerical results of load deflection curve for node a. (decreasing load)

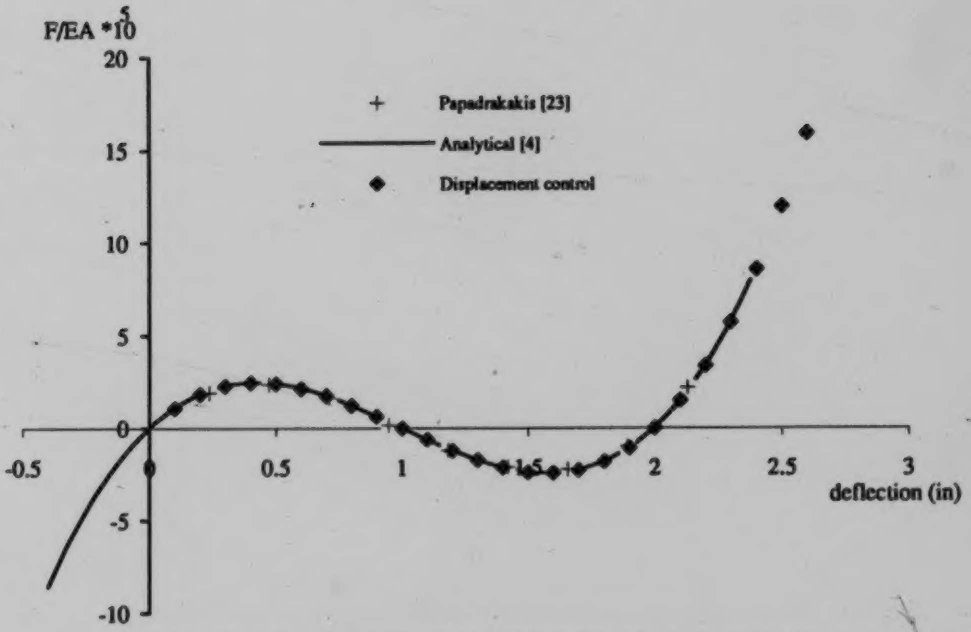


Figure (4.2e) Analytical and numerical results of load-deflection curve for node a .

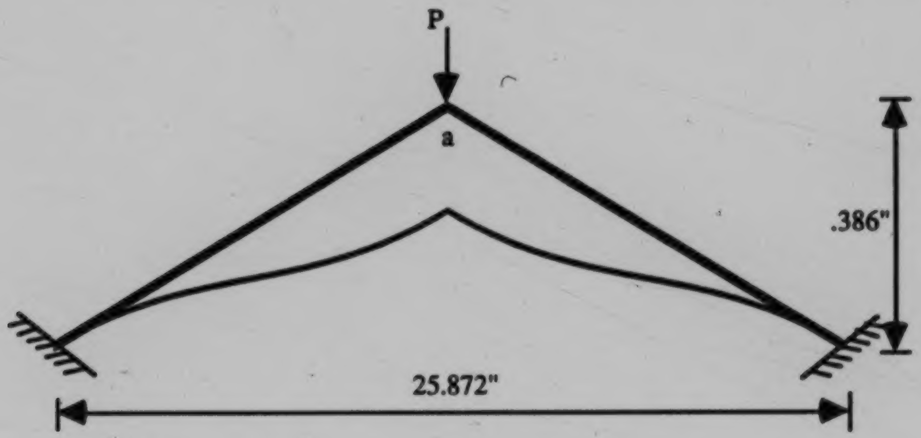
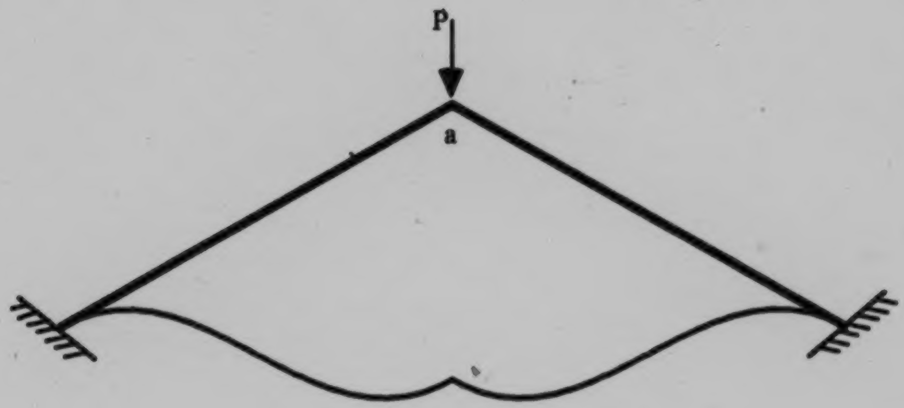


Figure (4-3): The two-member framewith the vertical force applied at rigid-jointed node a . (EA=1.855X10^6 lb, EI=9.27X10^3 lb/in^2)
(a) The initial and deformed configurations (before the snap through) .



(b) The initial and deformed configurations (after the snap through) .

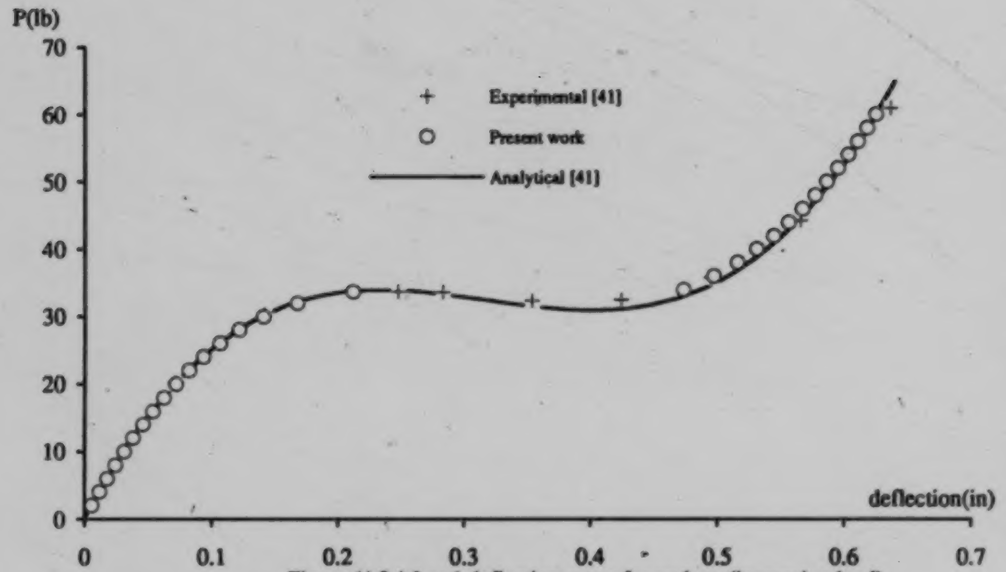


Figure (4.3c) Load-deflection curve for node a. (increasing load)

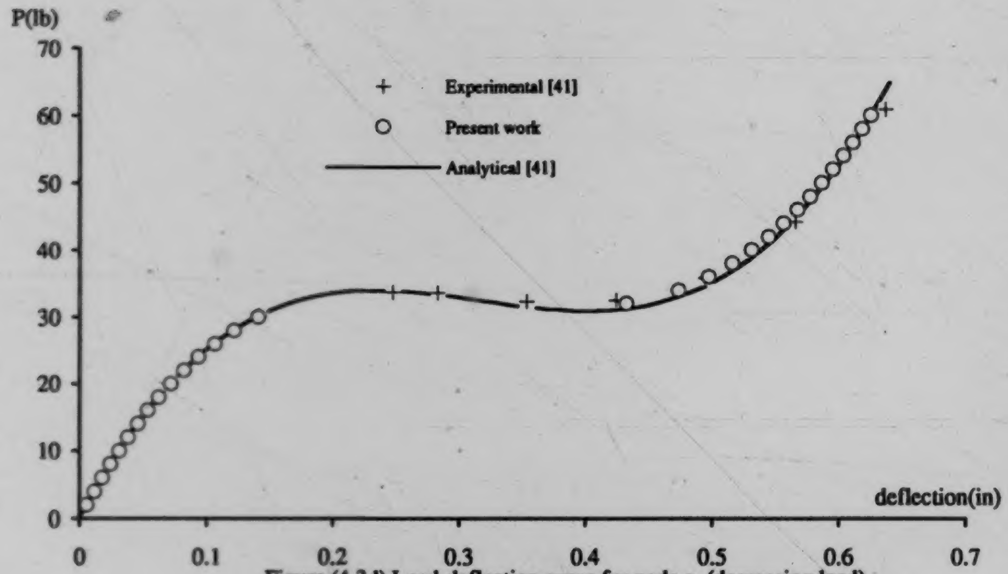


Figure (4.3d) Load-deflection curve for node a. (decreasing load)

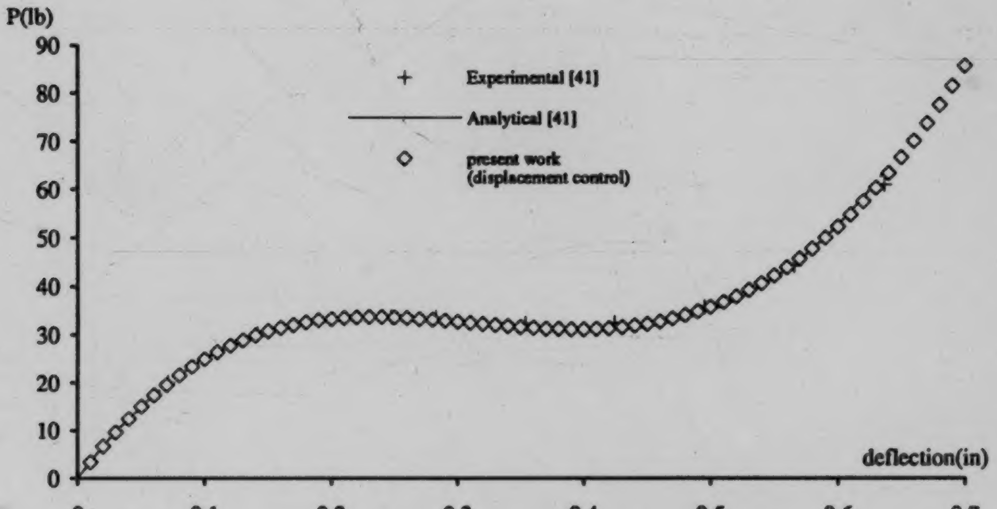


Figure (4.3e) Load-deflection curve for node a

Example (4-4): This structure shown in Figure (4.4a) has been analyzed by Powell and Simons [27]. They have adopted a displacement control strategy where a monotically increasing nodal displacement was chosen to increment the solution. The comparison of our results using load control with Powell's ones is given in Figure (4.4b).

Example (4-5): This structure shown in Figure (4.5a) has been analyzed by Meek and Tan [21] and also by Powell and Simons [27]. The comparison of our results with the previous ones for nodes 18, 26 (previous authors presented the results for these nodes only) is shown in Figure (4.5b). The results shown for after snap through are not published by other authors.

Example (4-6): This frame shown in Figure (4.6a) has been analyzed by Lipsett and Faulkner [18] using the segmental shooting technique by considering the rod as being comprised of a large number of segments, each of which experiences only small displacements so that a linear solution can be applied over each segment. The total nonlinear solution was obtained by assembling the segments together. Surana and Sorem [38] also used Finite Element method to analyze this problem. Our results (considering extensibility) using DR are compared in Figure (4.6c) with the results of [18] (with inextensible assumption). For a given force the midspan deflection would be higher considering extensibility comparing to inextensible one. In other word, to have a specific midspan deflection the required force would be less for the extensible case comparing to inextensible one which is reasonable.

Example (4-7): The behaviour of the structure shown in Figure (4.7a) (24 member hexagonal star-shaped shallow dome) was investigated by Paradiso *et al.* [25] where a secant-tangent approach is used with successive linear approximations. Also the results obtained by Papadrakakis [23] which used the beam-column approach to derive the nonlinear equilibrium equations and used DR procedure to solve them. Due to the symmetry in the initial configuration and loading the deformed configuration shown in Figure (4.7b) is also symmetric. Figure (4.7e) shows the load-deflection in the z direction for central node 1. For node 2 Figures (4.7f) and (4.7g) show the load-deflection in the z and x directions respectively. For both of these graphs in the beginning of applying the force both displacements are in positive direction. After a while to maintain the equilibrium the direction of force should be reversed.

Example (4-8): This structure shown in Figure (4.8b) has been analyzed by Paradiso *et al.* [25] where a secant-tangent approach is used with successive linear approximations. The deformed configuration after the first snap through is shown in Figure (4.8c). The deformed configurations before and after the second snap through are shown in Figure (4.8d) and Figure (4.8e) respectively. Due to the symmetry in the initial configuration and loading the deformed configurations in all different steps of loading are also symmetric. Figure (4.8f) shows the comparison of load-deflection curve of our results and the ones given in [25]

Example (4-9): For the structure shown in Figure (4.9b) with the side-view shown in Figure (4.9a) the deformed configurations after the first snap through, before the second snap through and after the second snap through are derived using present study (Figures (4.9c) through (4.9e)). In spite of being symmetric in the initial configuration, due to unsymmetric loading the deformed configurations in all different steps of loading are unsymmetric. The Figures (4.9f) and (4.9g) show the load-deflection curves for displacement in the z direction for nodes 1 and 2 respectively.

Example (4-10): For the structure shown in Figure (4.10a) the deformed configurations after the first snap through (due to node 1), Figure (4.10b), before the second snap through, Figure (4.10c), and after the second snap through (due to node 2), Figure (4.10d), are derived using present study. Figure (4.10e) shows the load-deflection curve for displacement in the y direction for node 1 during the first snap through. Figure (4.10f) shows the load-deflection curve for displacement in the x direction for node 2 during the first snap through. Figure (4.10g) shows the load-deflection curve for displacement in the y direction for node 1 during the second snap through. Figure (4.10h) shows the load-deflection curve for displacement in the x direction for node 2 during the second snap through.

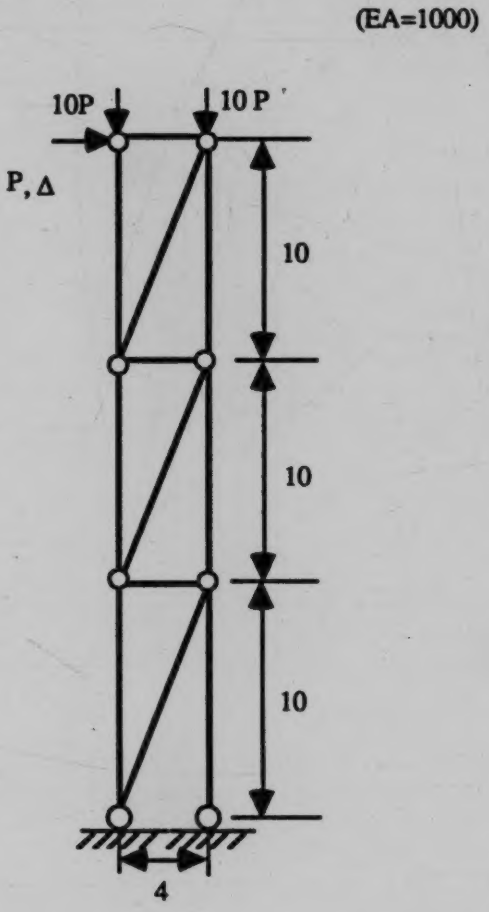

(EA=1000)
10P
10 P
P, Δ
10
10
10
4

Figure (4-4): (a) The geometry of tower truss with the combined vertical and horizontal forces.

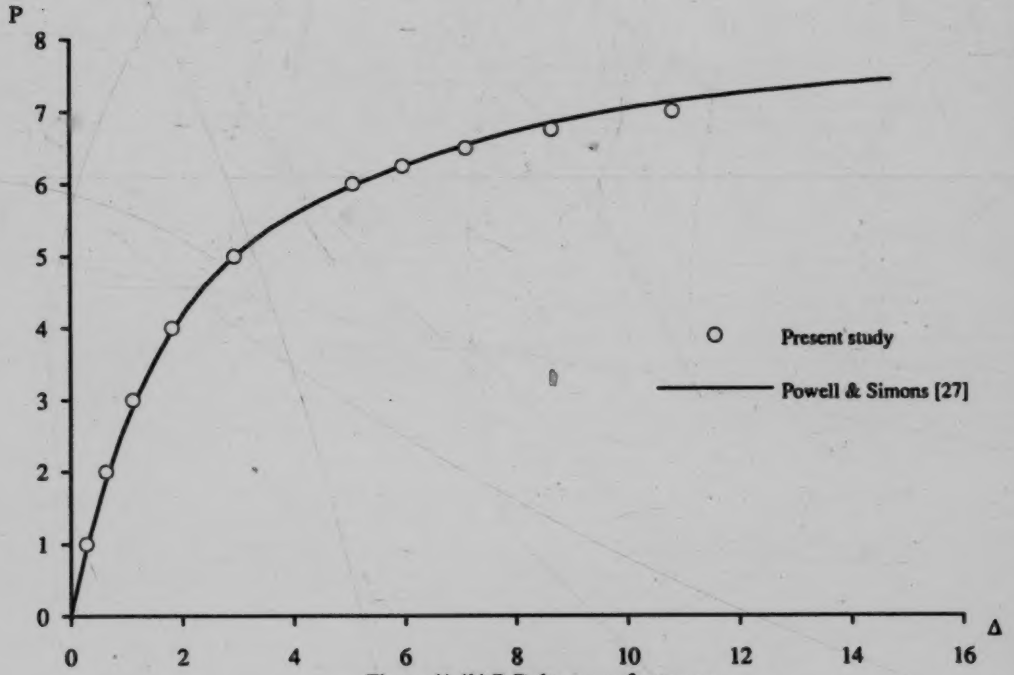


Figure (4.4b) P-Delta curve for tower

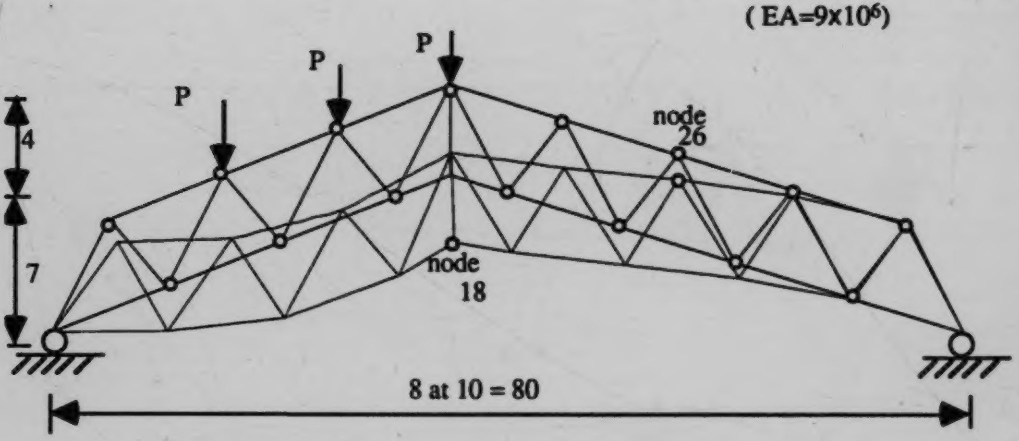

(EA=9x10^6)
P
P
P
4
7
node
26
node
18
8 at 10 = 80

Figure (4-5): The truss-arch with the vertical forces applied at pin-jointed nodes.
(a) The initial and deformed configurations .

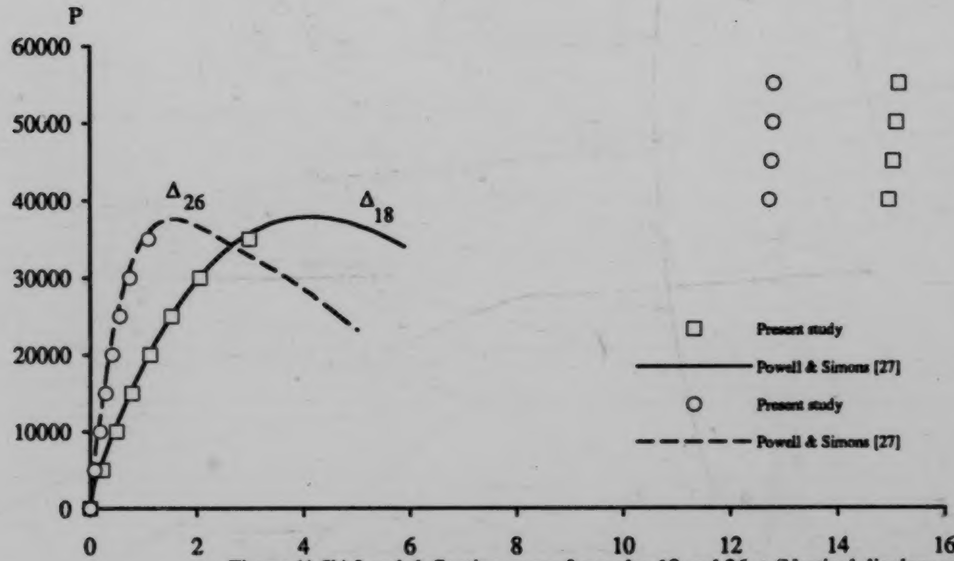


Figure (4.5b) Load-deflection curve for nodes 18 and 26 Δ(Vertical displacement)

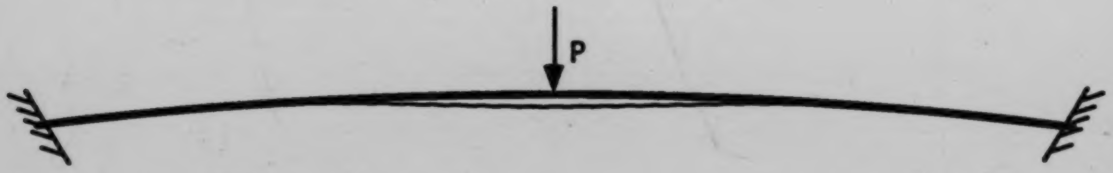
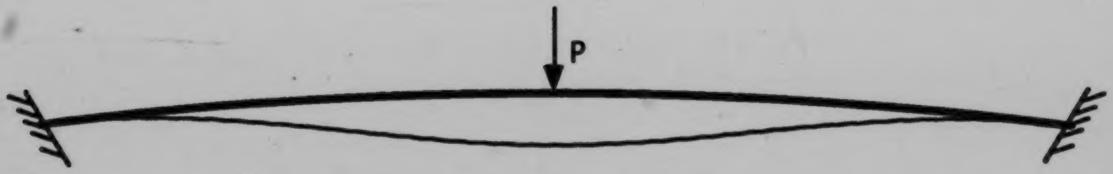


Figure (4-6): The Fixed-fixed shallow circular arch with midspan concentrated vertical load.
(EI=5493 lb.in^2, EA=1875000 lb, Radius=133.114 in, Span angle=14.67458°)

(a) The initial and deformed configurations (before the snap through).



(b) The initial and deformed configurations (after the snap through).

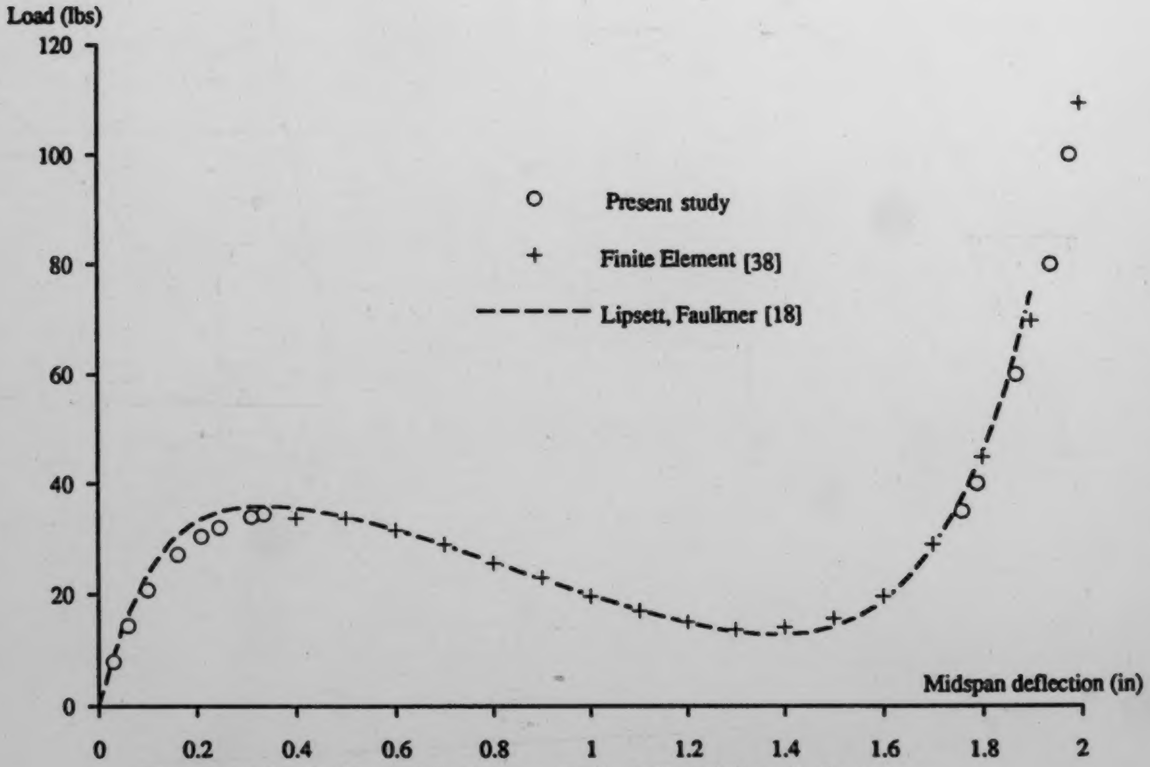


Figure (4.6c) Load-midspan deflection curve for fixed-fixed shallow circular arch

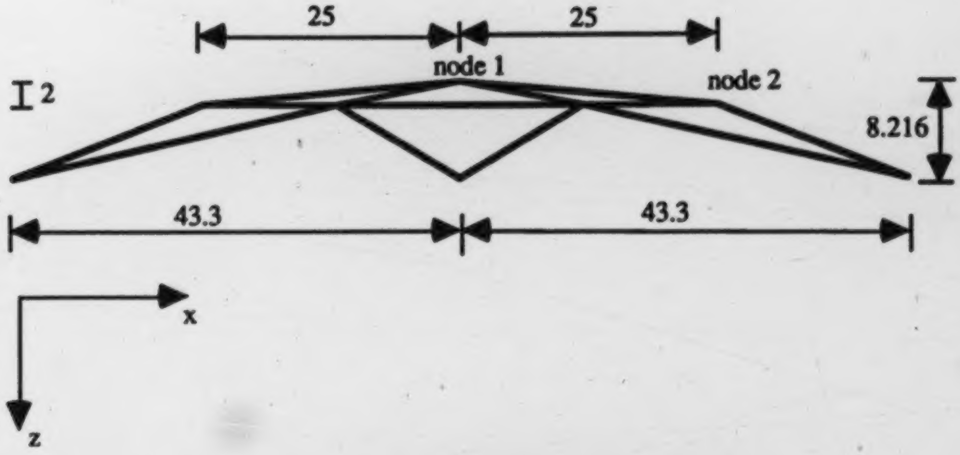

25
25
node 1
node 2
2
8.216
43.3
43.3
x
z

Figure (4.7): The space structure with ball-jointed nodes, the vertical force applied at centeral node 1. (all dimensions are in cm)
(a) The side-view of the initial configuration.

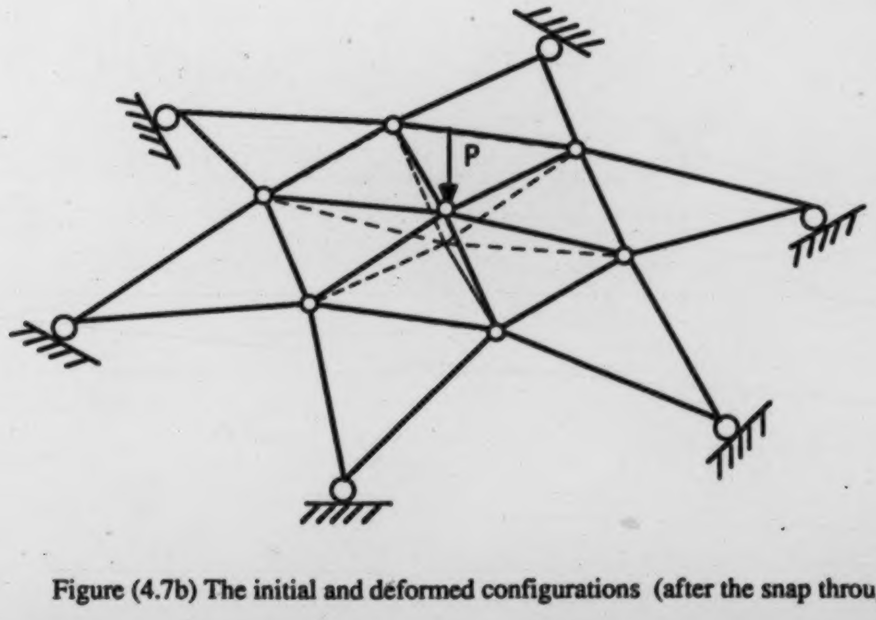


Figure (4.7b) The initial and deformed configurations (after the snap through) .

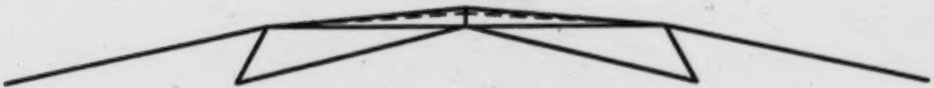


Figure (4.7c) The side-view of the initial and deformed configurations (before the snap through).

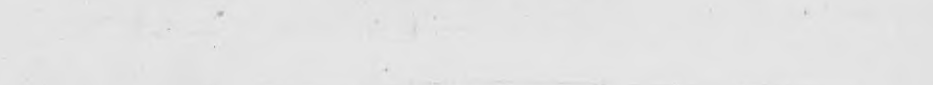


Figure (4.7d) The side-view of the initial and deformed configurations (after the snap through).

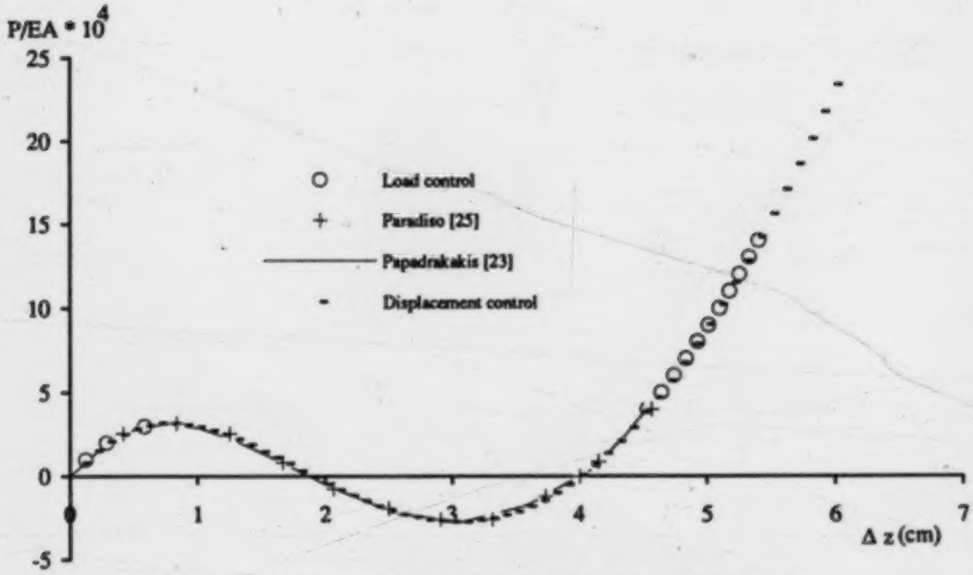


Figure (4-7c) Load-deflection curve for node 1

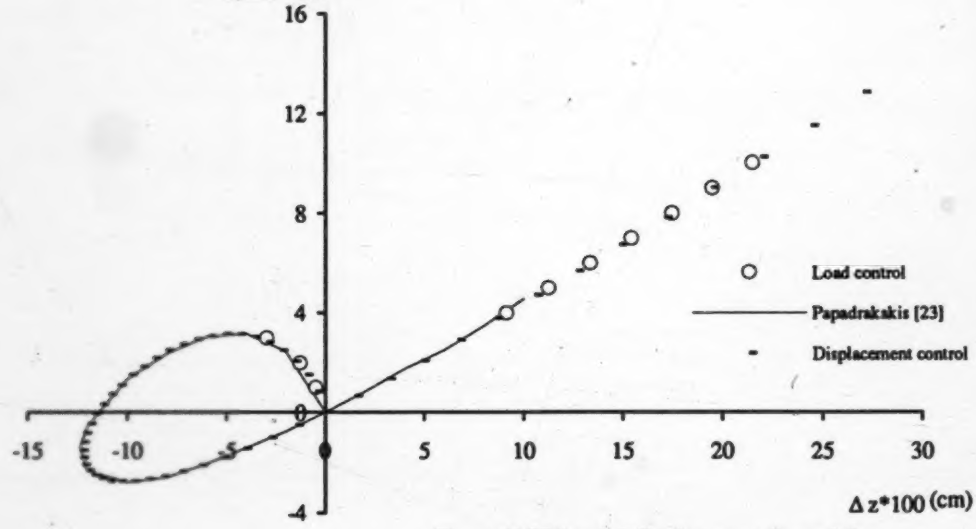


Figure (4.7f) Load-deflection curve for node2

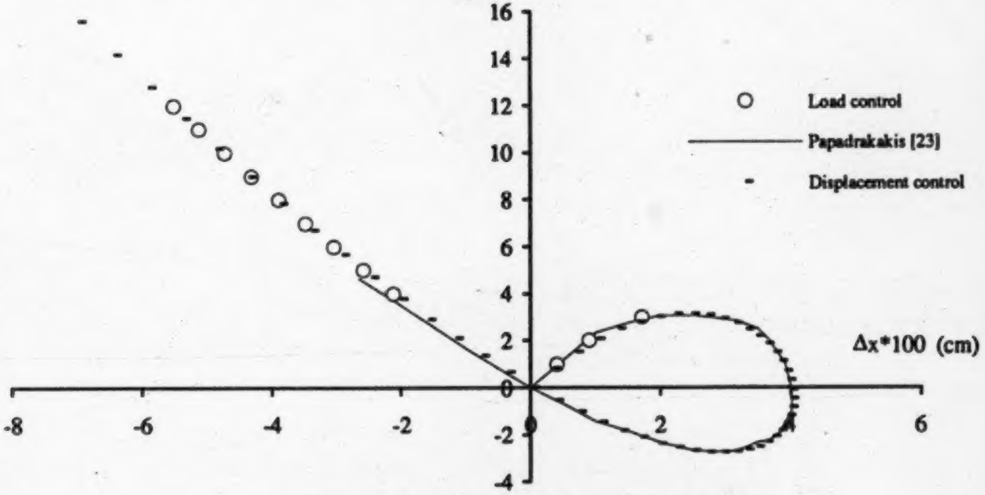


Figure (4-7g) Load-deflection curve for node 2

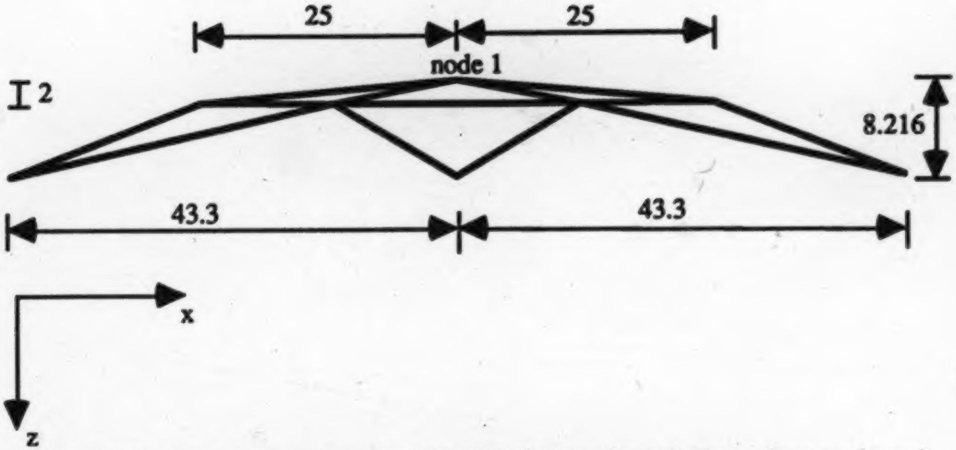

25
25
node 1
2
8.216
43.3
43.3
x
z

Figure (4.8): The space structure with ball-jointed nodes (all dimensions are in cm).

(a) The side-view of the initial configuration.

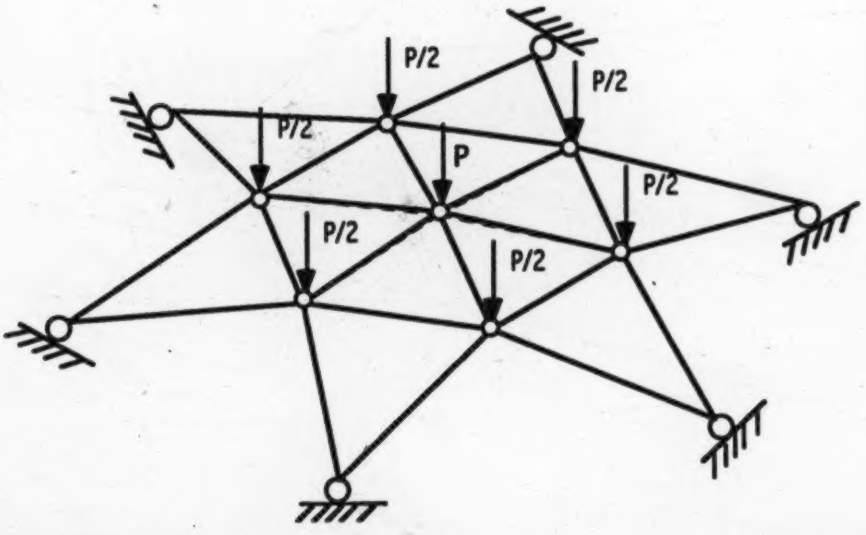

P/2
P/2
P/2
P
P/2
P/2
P/2

Figure (4.8b) The initial and deformed configurations (before the first snap through).

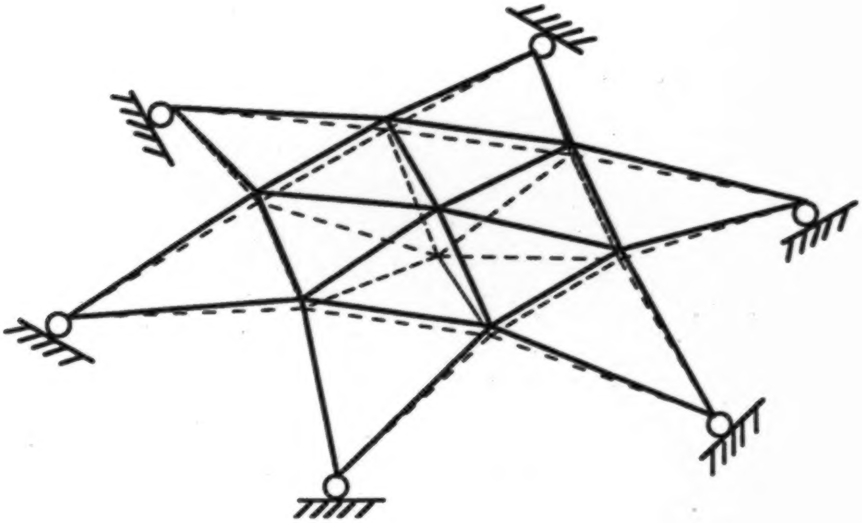


Figure (4.8c) The initial and deformed configurations (after the first snap through).

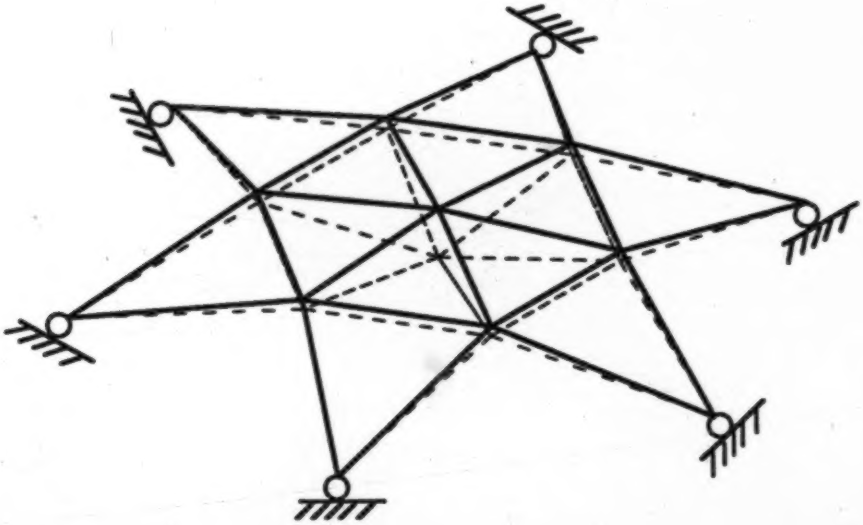


Figure (4.8d) The initial and deformed configurations (before the second snap through).

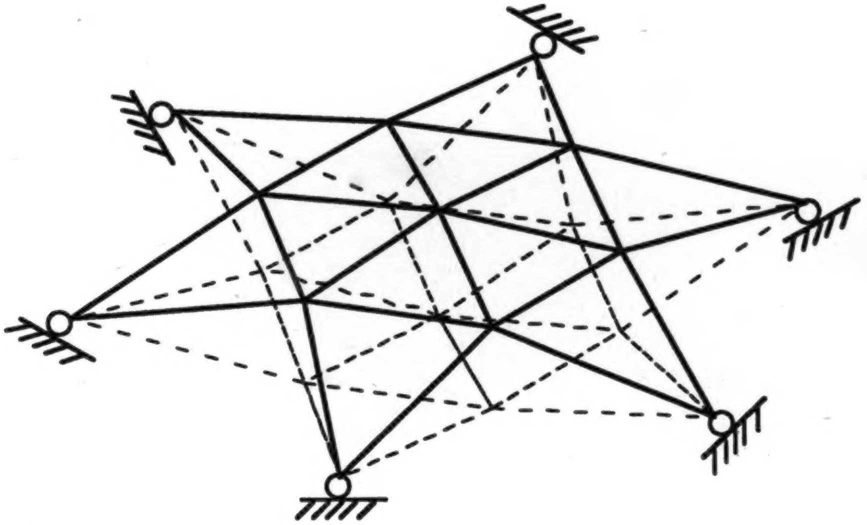


Figure (4.8e) The initial and deformed configurations (after the second snap through) .

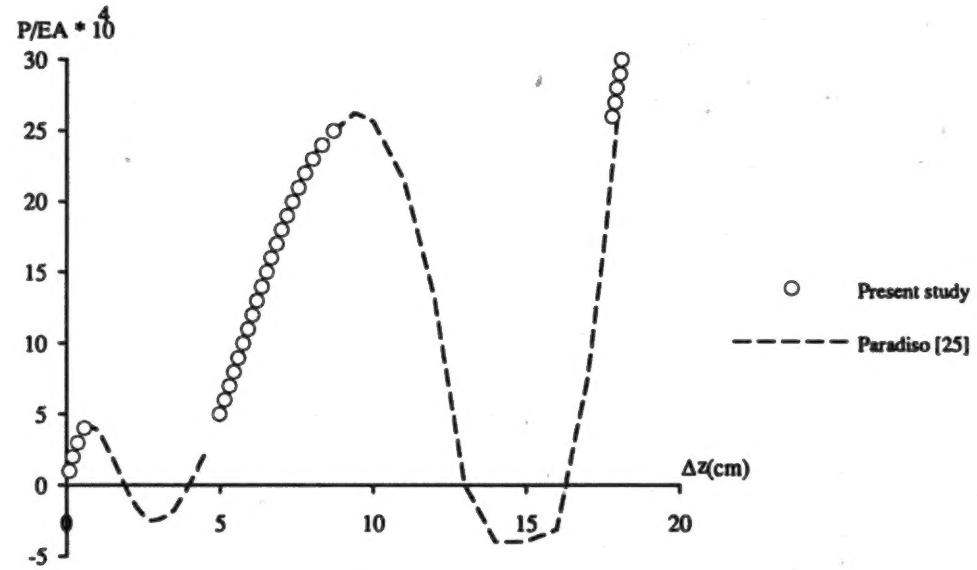


Figure (4.8f) Load-deflection curve for node 1

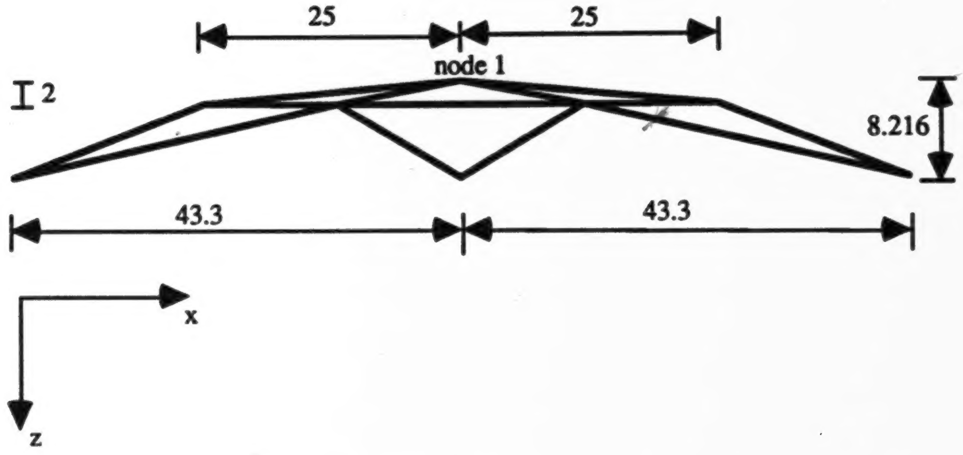

25
25
node 1
2
8.216
43.3
43.3
x
z

Figure (4.9): The space structure with ball-jointed nodes (all dimensions are in cm).

(a) The side-view of the initial configuration.

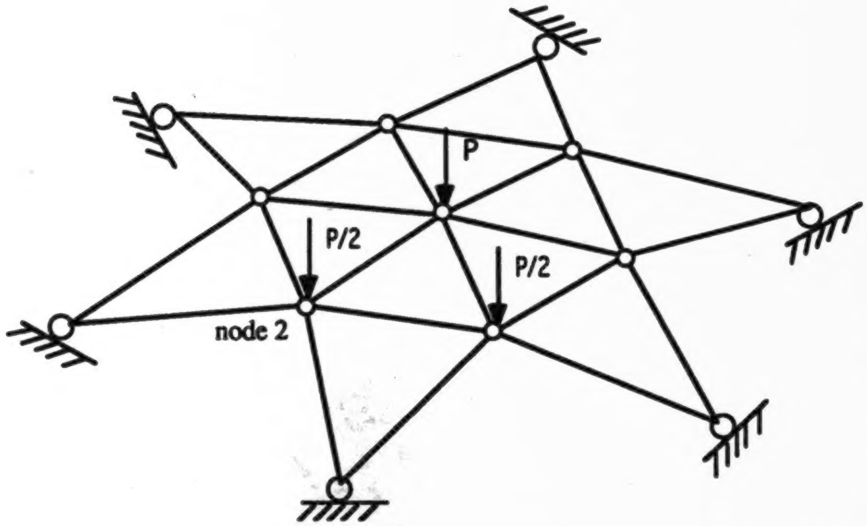

P
P/2
P/2
node 2

Figure (4.9b) The initial and deformed configurations (before the first snap through) .

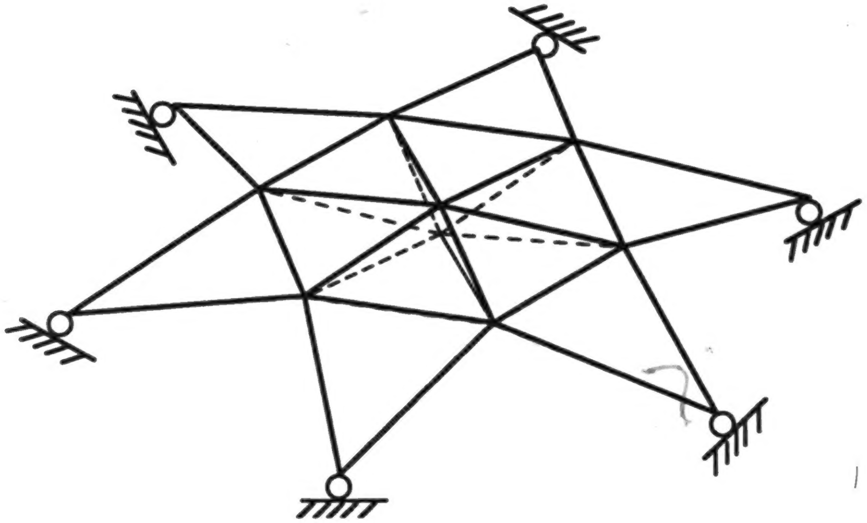


Figure (4.9c) The initial and deformed configurations (after the first snap through).

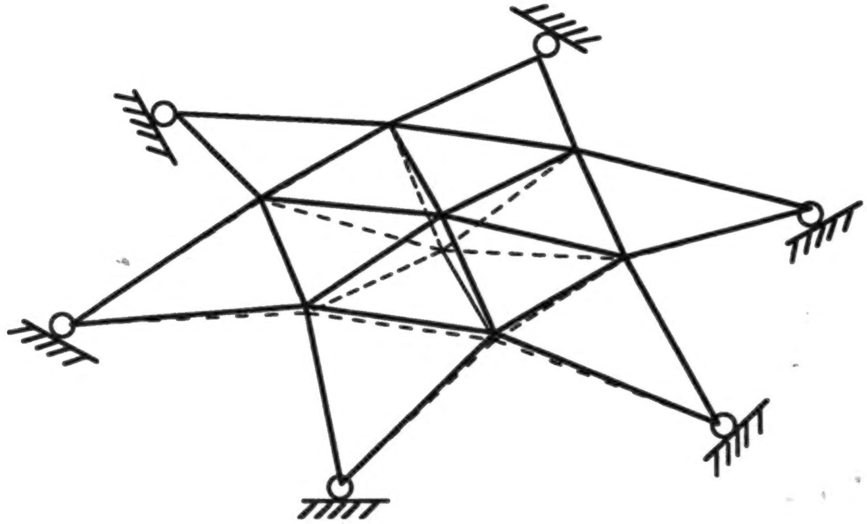


Figure (4.9d) The initial and deformed configurations (before the second snap through).

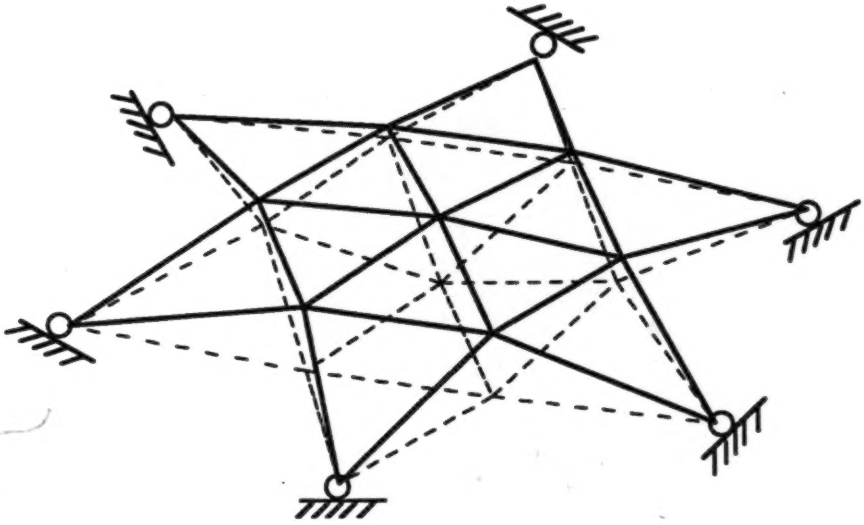


Figure (4.9e) The initial and deformed configurations (after the second snap through).

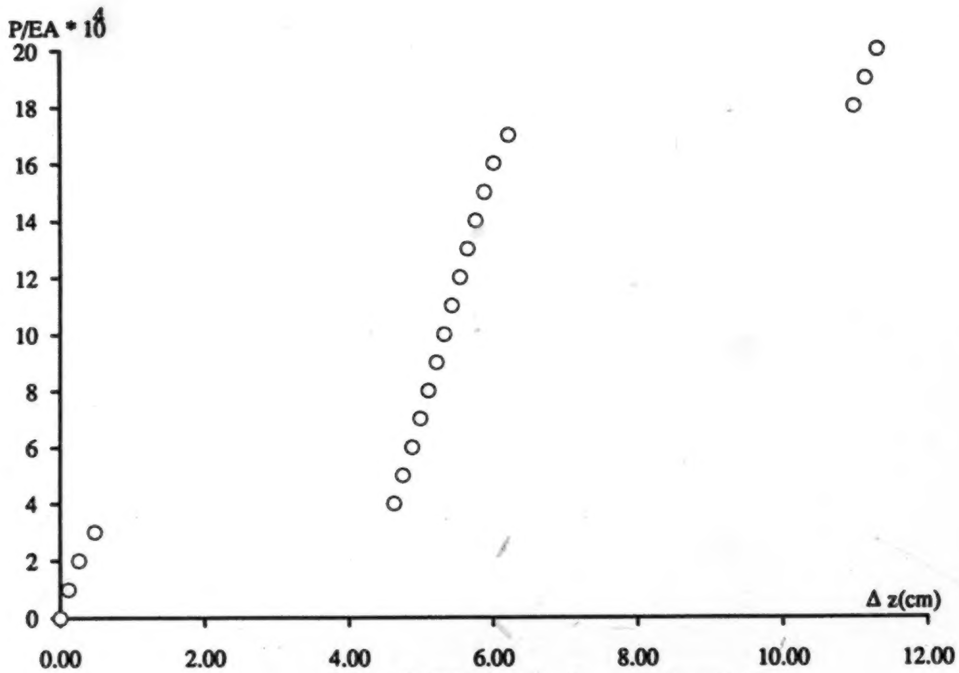


Figure (4.9f) Load-deflection curve for node 1

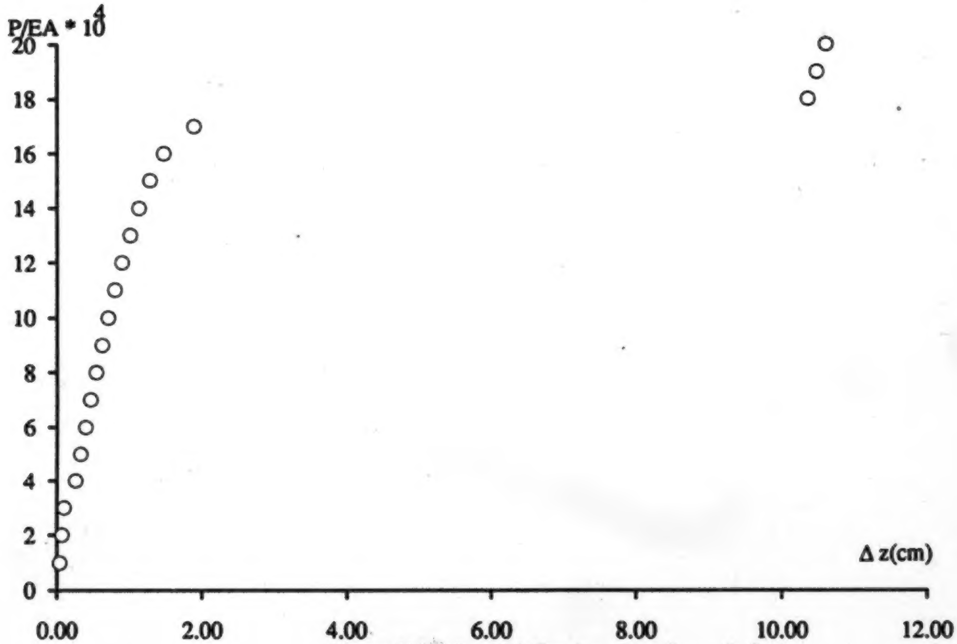


Figure (4.9g) Load-deflection curve for node 2

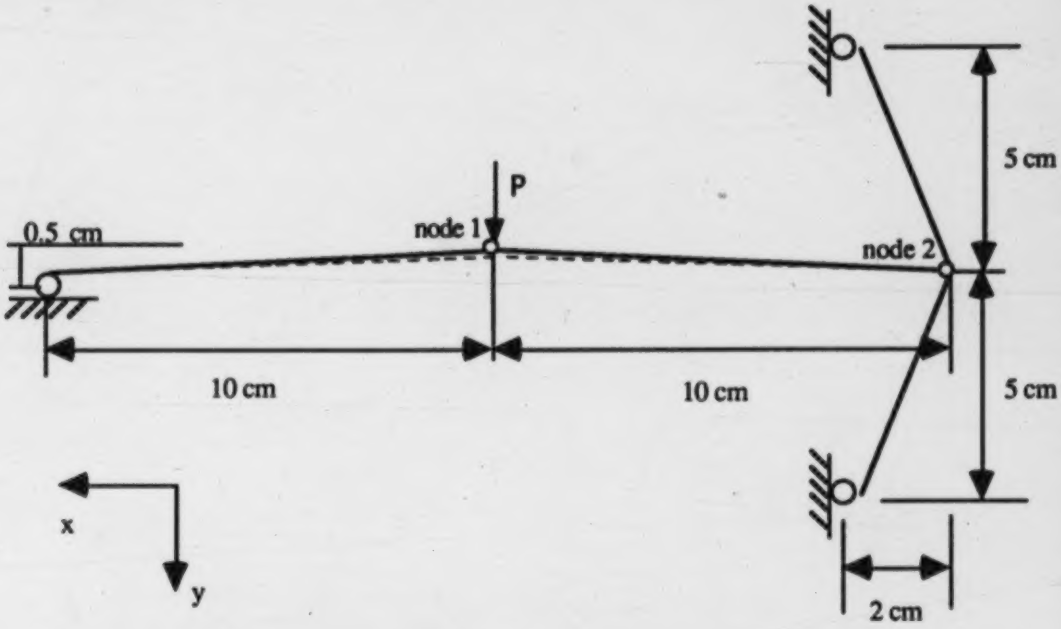
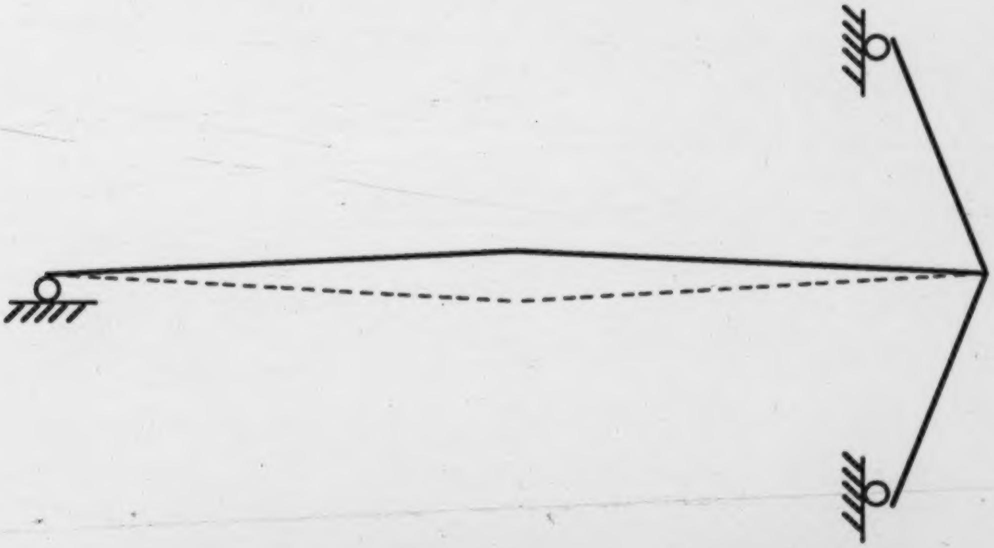
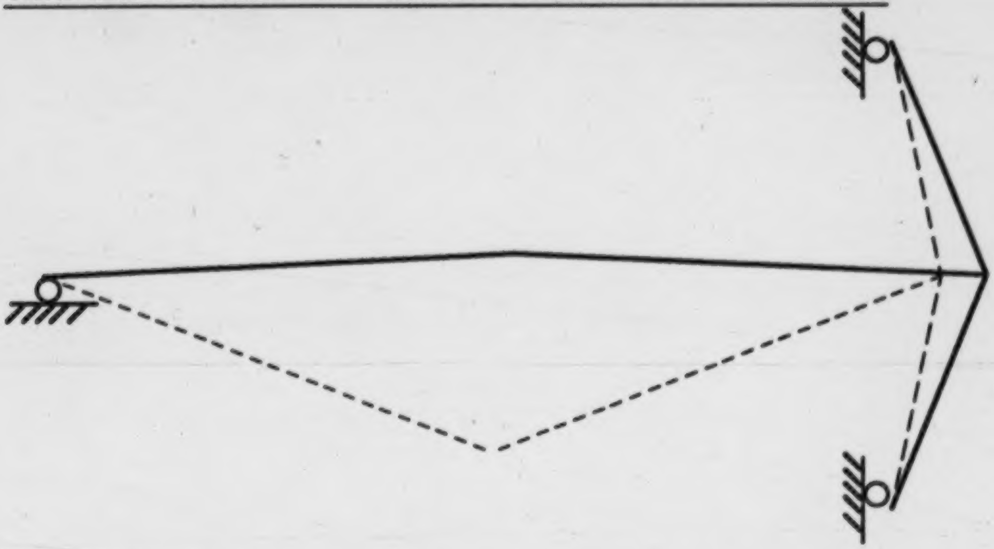

5 cm
P
0.5 cm
node 1
node 2
10 cm
10 cm
5 cm
x
y
2 cm

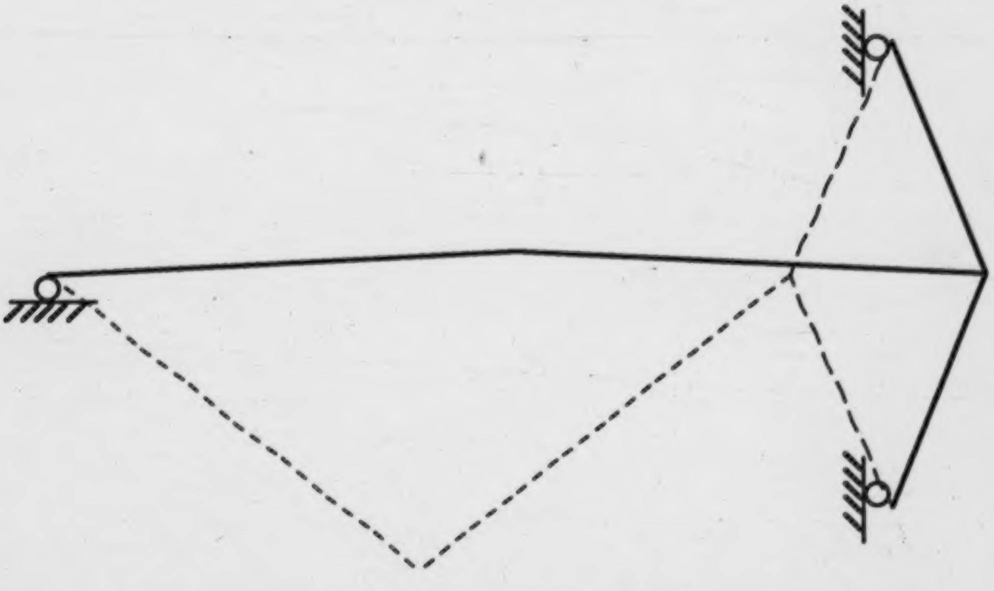
Figure (4-10): The planar structure with ball-jointed nodes .
(a) The initial and deformed configurations (before the first snap through) .



(b) The initial and deformed configurations (after the first snap through) .



(c) The initial and deformed configurations (before the second snap through) .



(d) The initial and deformed configurations (after the second snap through) .

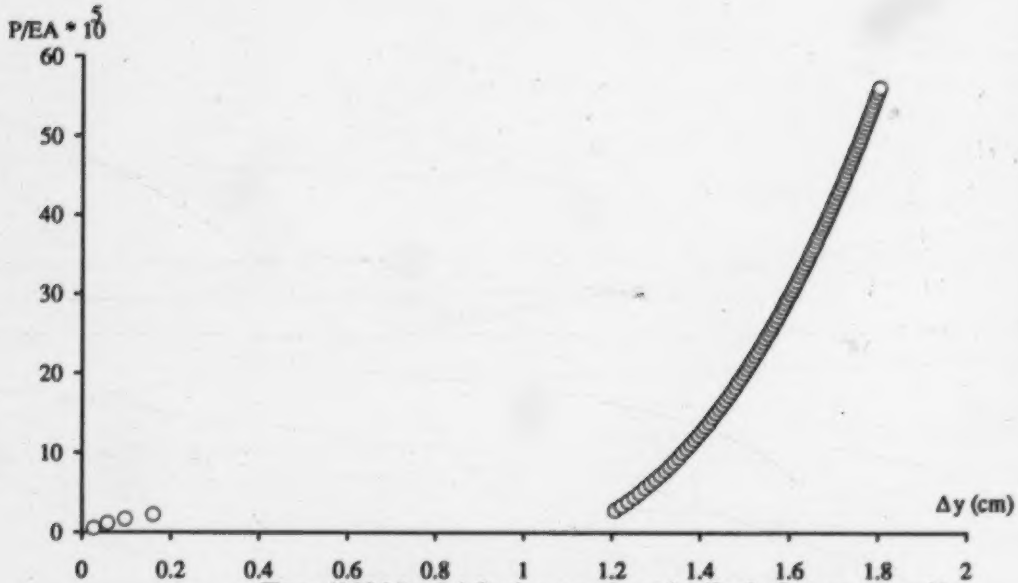


Figure (4.10e) Load-deflection curve for node1 (for the first snap through).

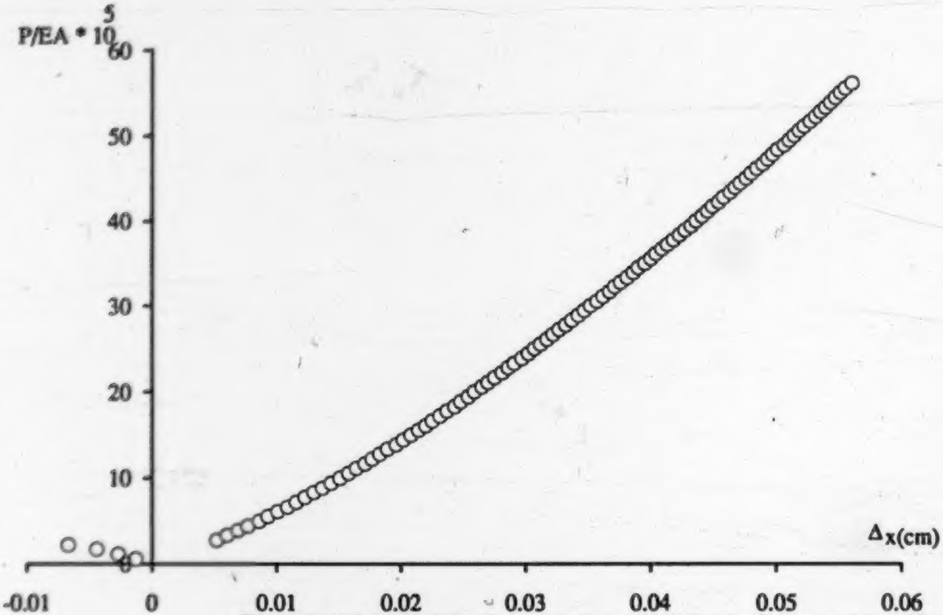


Figure (4.10f) Load-deflection curve for node 2 (for the first snap through).

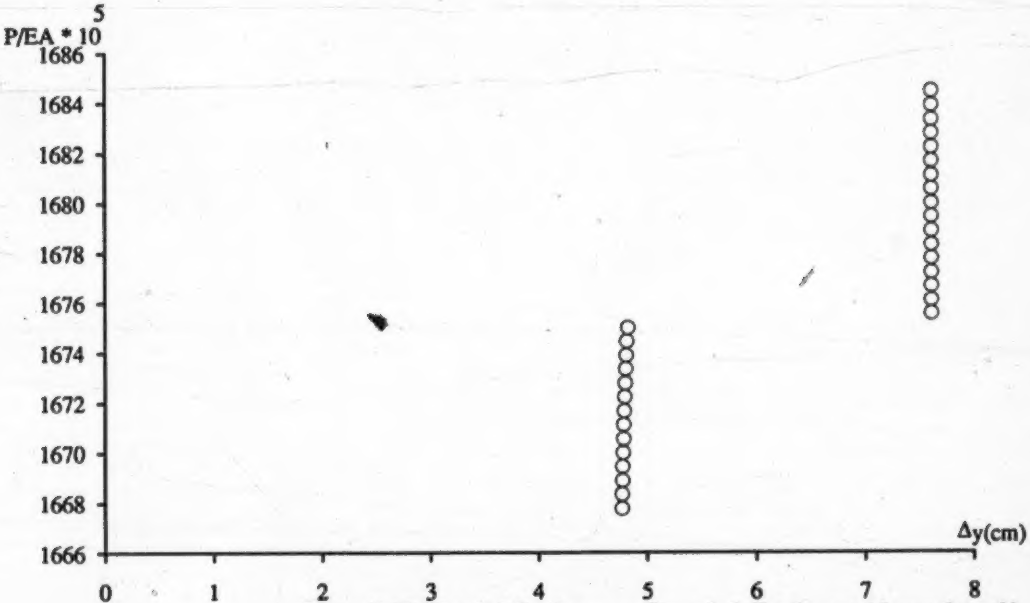


Figure (4.10g) Load-deflection curve for node 1 (for the second snap through).

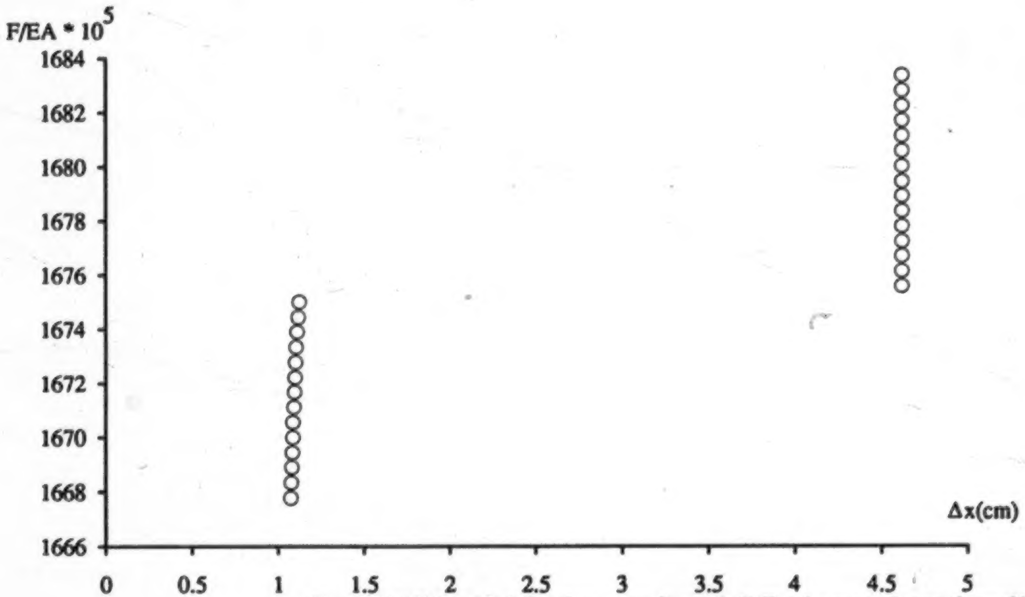


Figure (4.10h) Load-deflection curve for node 2 (for the second snap through).

Example (4-11): For the structure shown in Figure (4.11a) increasing the load makes the first snap through (due to node 1). The deformed configuration after the first snap through is shown in Figure (4.11b). Increasing the load forms the second snap through (due to node 4). Deformed configurations before and after the second snap through are shown in Figures (4.11c) and (4.11d) respectively which are derived using present study. Figures (4.11e) and (4.11f) show the load-deflection curves for displacement in the y direction for node 1 during the first and the second snap through respectively. Figure (4.11g) shows the load-deflection curve for displacement in the x direction for node 2 during the second snap through. We see that before the second snap through moves to the right and at the second snap through there is a little jump to the left and then starts to move to the right. The same thing happens to the vertical displacement for node 3 (Figure (4.11h)). Figure (4.11i) shows the load-deflection curve for displacement in the x direction for node 5 during the second snap through. Figure (4.11j) shows the load-deflection curve for displacement in the y direction for node 4 during the second snap through.

Example (4-12): For the space structure shown in Figure (4.12a) the material properties are $EA = 1.855\text{x}10^6\ lb$, $EI = 9.27\text{x}10^3\ lb/in^2$. Due to the symmetry in the initial configuration and loading only bending is involved without any twist and also the deformed configuration (dashed-line) shown in Figure (4.7a) is symmetric. The deformed configuration after snap through is derived using DR. The load-deflection curve for displacement in the z direction for node a for both techniques load control and displacement control are given in Figure (4.12).

Example (4-13): The initially curved cantilever shown in Figure (4.13), loaded out of the initial plane of curvature by a dead load is considered here. The material properties are $E = 10^7\ psi$, $G = 3.33\text{x}10^6\ psi$. The cross section of beam is 1x1 *in* and the radius of circle in the initial configuration is 100 *in*. Raboud *et al.* [29] also studied this problem considering inextensible rod theory. Our results using Newton-Raphson are compared with the results of [29] in table (4.1). This table shows the predicted coordinates of the tip of the cantilever for various dimensionless loads k where $k = \frac{pR^2}{EI}$.

Example (4-14): The deformed configuration of a circular beam shown in Figure (4.14) with one end clamped and the other end free with applied force at free end is derived using NR. This example is considered here for the first time. The material properties are $E = 10^7$ *psi*, $G = 3.33\text{x}10^6$ *psi*. The cross section of beam is 1x1 *in* and the radius of circle in the initial configuration is 100 *in*. In initial configuration the coordinate of free end is (0,100,0). After applying dead load $P = 32$ *lb* the coordinate of free end becomes $27.439'', 22.71'', 336.5''$. Also applying dead load $P = 8$ *lb* the coordinate of free end becomes $4.55'', 71.37'', 136.95''$.

Example (4-15): The buckling of a cantilever beam shown in Figure (4.15) is derived here. The material properties are $E = 20000$ *psi*, $A = 42$ in^2 , $I = 6482$ in^4, $Length = 400$ *in*. Using Eulerian equation for a cantilever beam $p_{cr} = \frac{\pi^2 EI}{4l^2}$ gives $p_{cr} = 1999.2lb$. Numerical result using NR gives $p_{cr} = 2050lb$ which is quite close to the theory($p_{cr} = 1999.2lb$).

Example (4-16): The deformed configuration of initially curved cantilever beam shown in Figure (4.16) is derived using both NR and DR. The material properties are $EI = 8.33\text{x}10^5$ $lb.in^2$, $GJ = 5.5\text{x}10^5$ $lb.in^2$, $EA = 100$ *lb*. The radius of arch in the initial configuration is 100 *in*. Table (4.2) shows the numerical results for both methods and the DR result is compared to NR.

In the following examples (considered here for the first time) we apply a prescribed displacement and we derive the deformed configuration using DR. For the prescribed displacement for a specific node, we usually apply it step by step, use DR to derive the displacements for the rest of the structure at that given step of specified displacement, then take the deformed configuration in that step as the initial condition for the next step and continue it until we reach to the desired specified displacement for that specified node. The external required forces and moments for equilibrium would be the negative of resultant of the internal forces and moments at that specified node. To solve these problems with applying load we should restart the problem by applying the external required forces and moments at the specific node and derive the deformed configuration.

Example (4-17): The nonlinear deformed configuration of a fixed-fixed semicircle shown in Figure (4.17) under prescribed displacement in the middle is derived using DR method. The material properties are $EI = 8.33\text{x}10^5$ $lb.in^2$, $GJ = 5.5\text{x}10^5$ $lb.in^2$, $EA = 100$ lb. The radius of semicircle in the initial configuration is 100 in. The prescribed displacement in the middle is 34 in. Due to the symmetry in the initial configuration and symmetric prescribed displacement the deformed configuration shown in Figure (4.17) is also symmetric.

Example (4-18): The nonlinear deformed configuration of a structure shown in Figure (4.18) with two circular arcs (two 1/8 of a circle) with fixed-fixed ends under prescribed displacement in the middle is derived using DR method. The material properties are $EI = 8.33\text{x}10^5$ $lb.in^2$, $GJ = 5.5\text{x}10^5$ $lb.in^2$, $EA = 100$ lb. The radius of arc in the initial configuration is 100 in. The prescribed displacement in the middle is 4.7 in. Due to the symmetry in the initial configuration and and symmetric prescribed displacement the deformed configuration shown in Figure (4.17) is also symmetric.

Example (4-19): The nonlinear deformed configuration of a fixed-free semicircle shown in Figure (4.19) under prescribed displacement at the free end is derived using DR method. The material properties are $EI = 8.33\text{x}10^5$ $lb.in^2$, $GJ = 5.5\text{x}10^5$ $lb.in^2$, $EA = 100$ lb. The radius of semicircle in the initial configuration is 100 in. The prescribed displacement in the middle is 56.5 in.

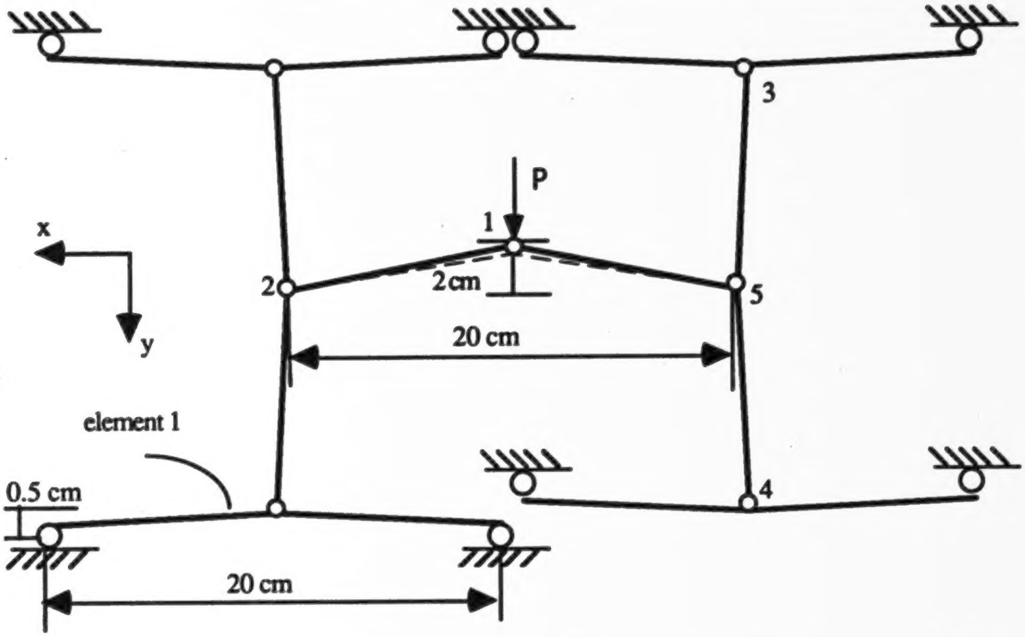
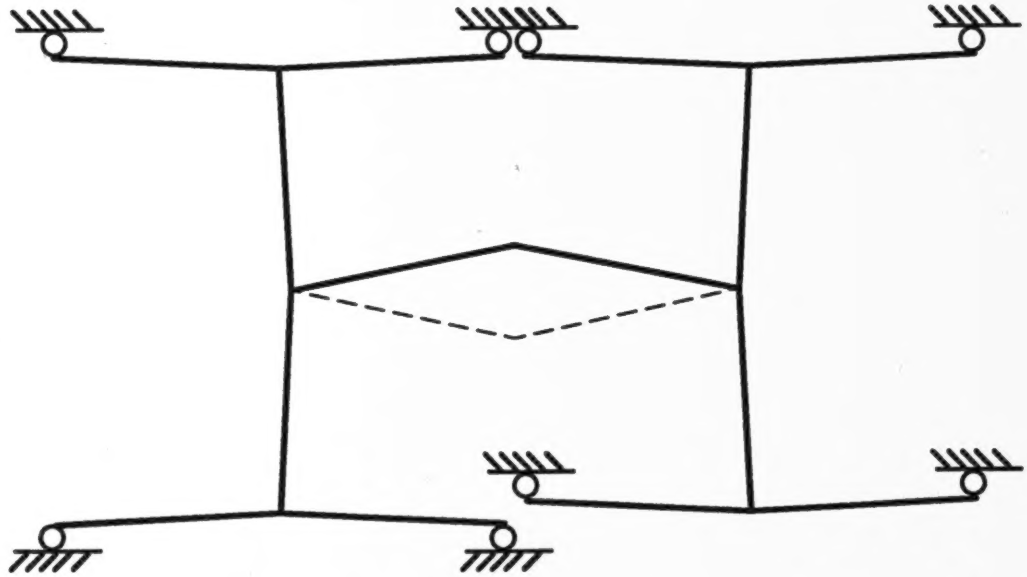
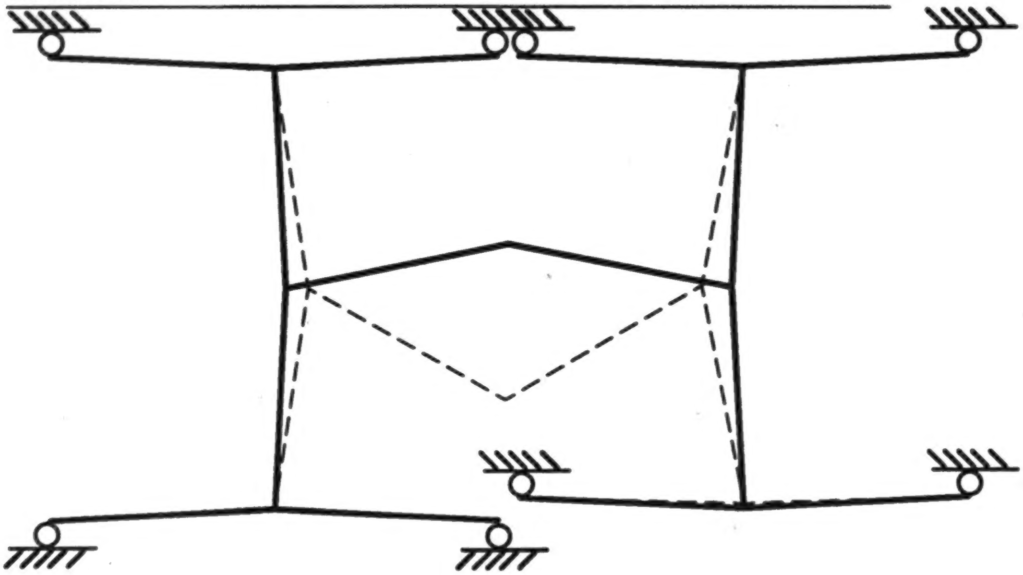


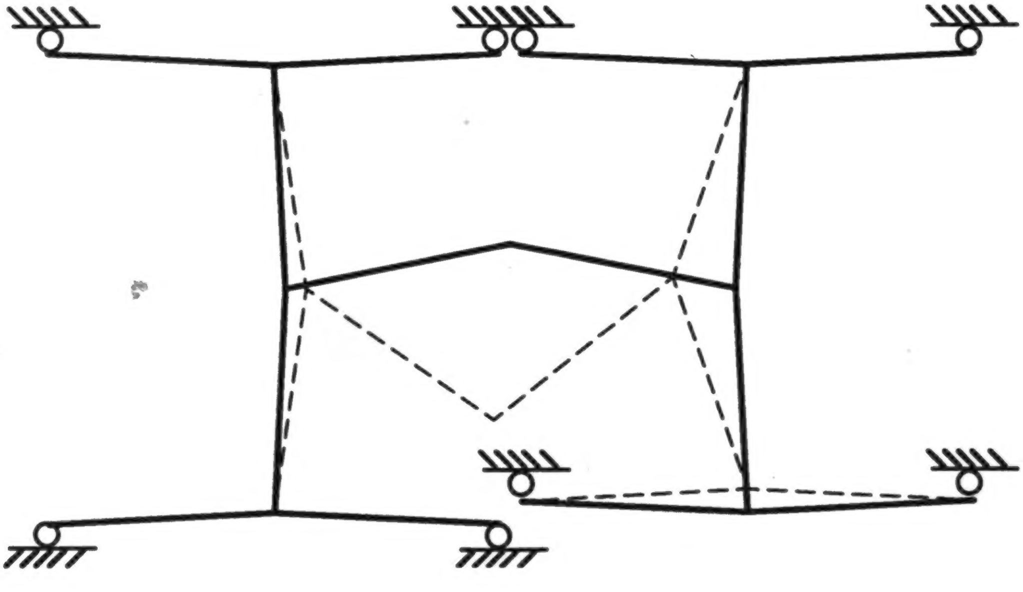
Figure (4-11): The planar structure with ball-jointed nodes . The dimensions of all the elements are similar to the element 1 except the middle one which is given.
(a) The initial and deformed configurations (before the first snap through) .



(b) The initial and deformed configurations (after the first snap through) .



(c) The initial and deformed configurations (before the second snap through).



(d) The initial and deformed configurations (after the second snap through).

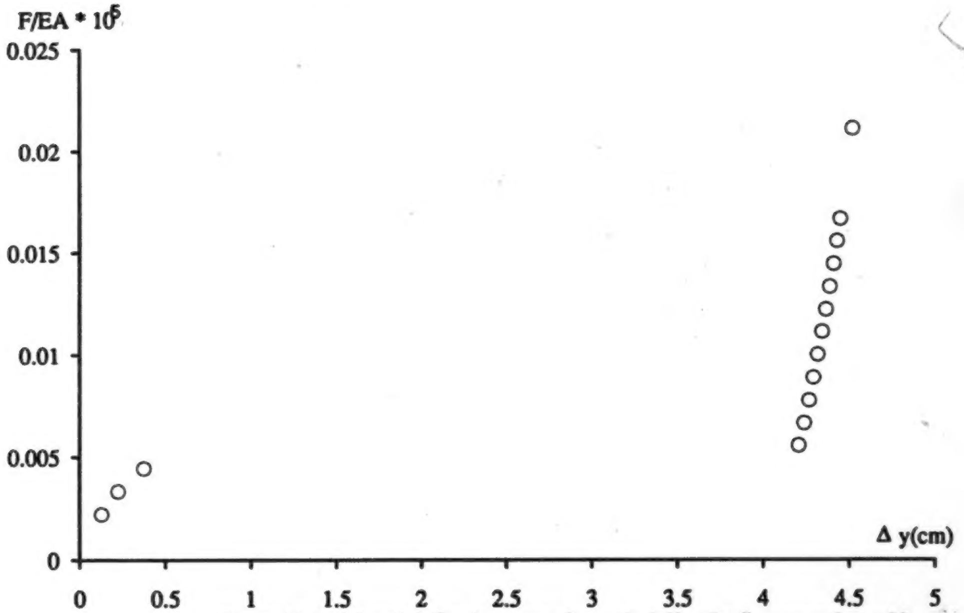


Figure (4.11e) Load-deflection curve for node 1 (for the first snap through).

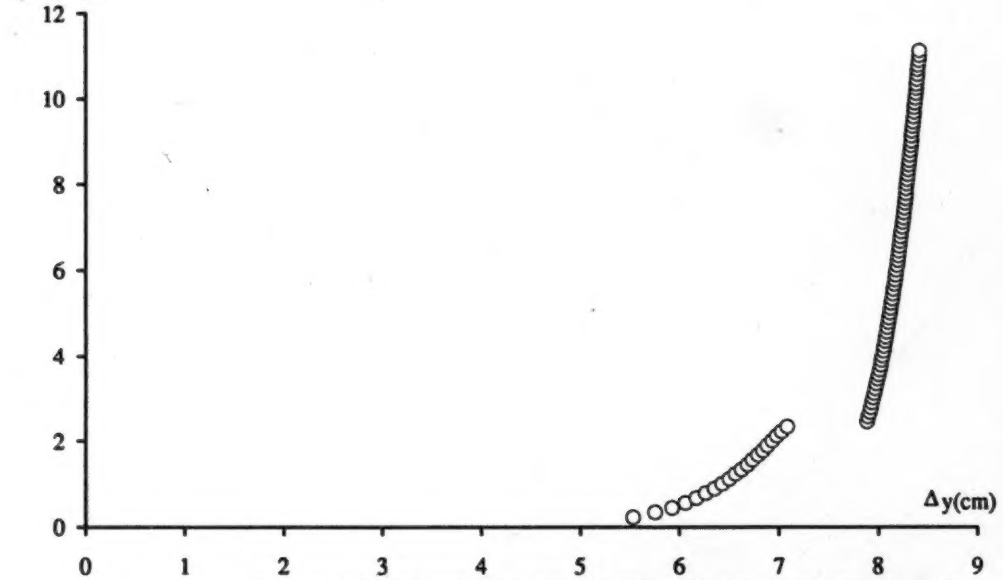


Figure (4.11f) Load-deflection curve for node 1 (for the second snap through).

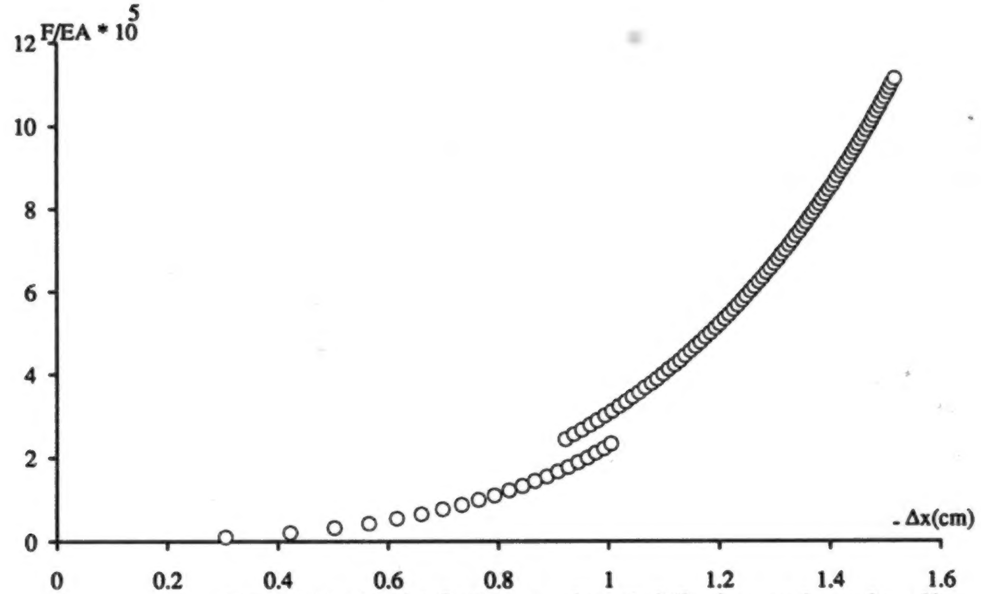


Figure (4.11g) Load-deflection curve for node 2 (for the second snap through).

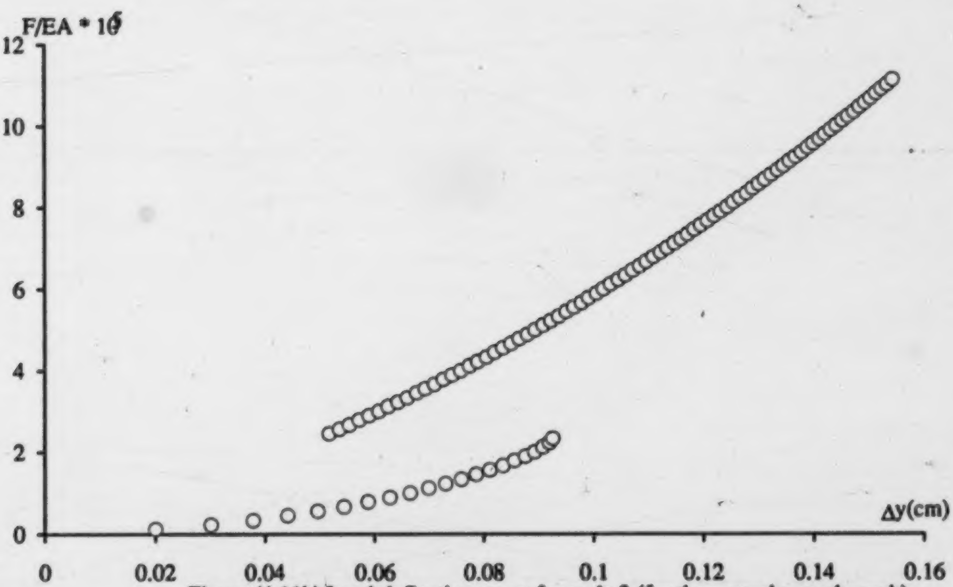


Figure (4.11h) Load-deflection curve for node 3 (for the second snap through).

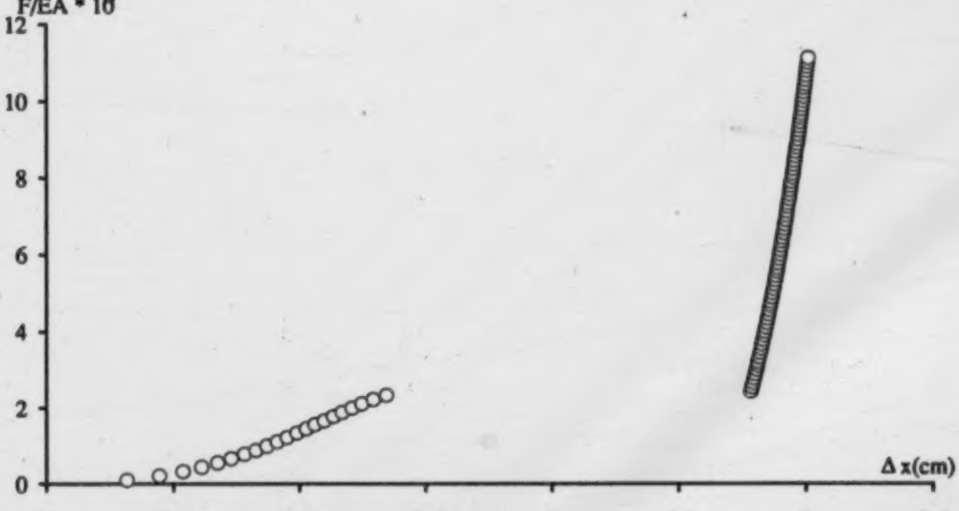


Figure (4.11i) Load-deflection curve for node 5 (for the second snap through)

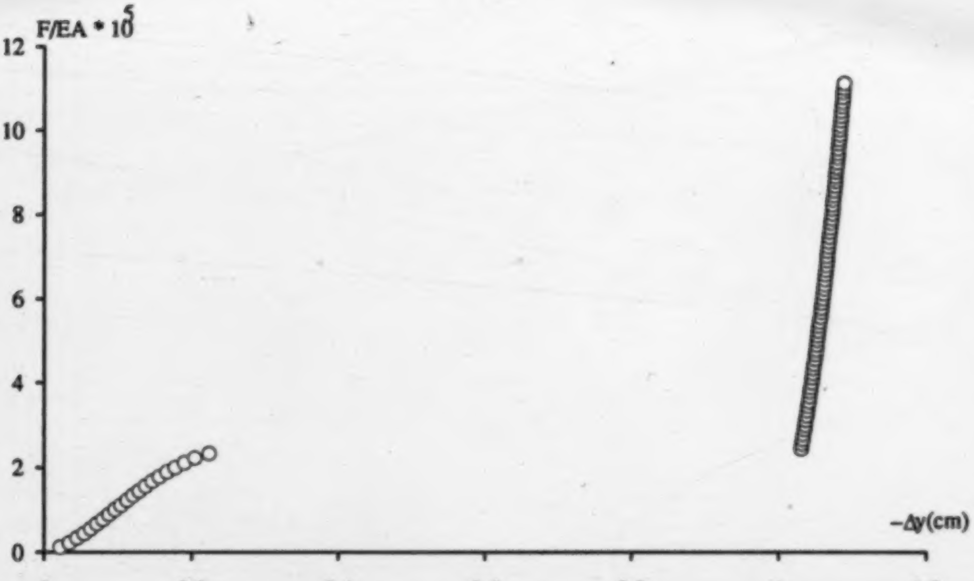


Figure (4.11j) Load-deflection curve for node 4 (for the second snap through).

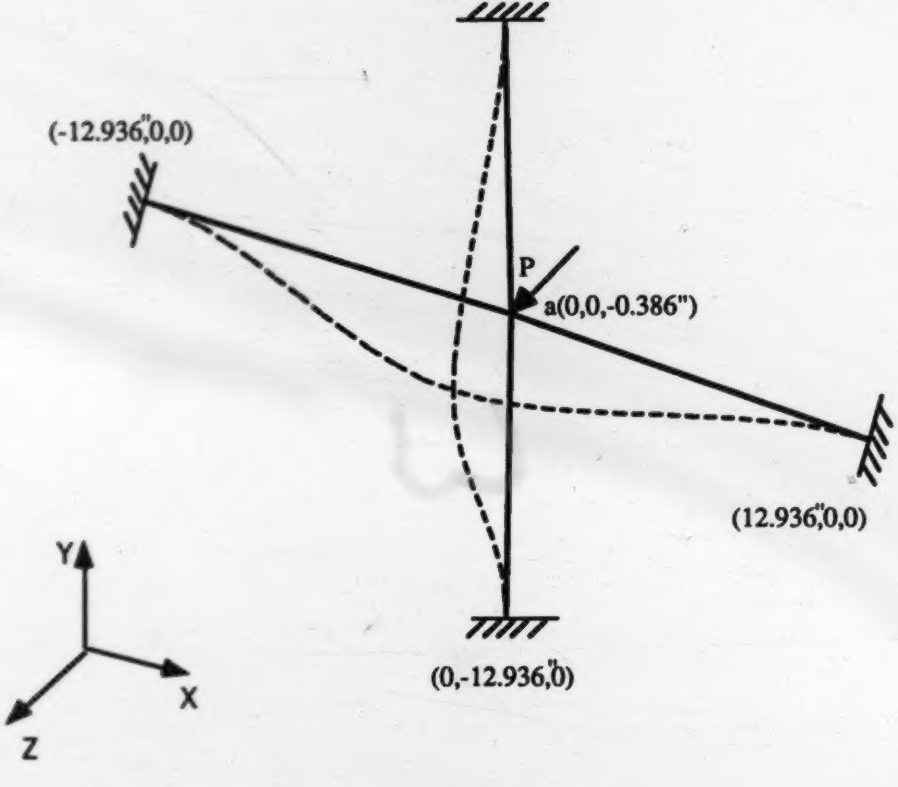

(0,12.936",0)
(-12.936",0,0)
P
a(0,0,-0.386")
(12.936",0,0)
(0,-12.936",0)
Y
X
Z

Figure (4-12): The space frame with the applied force at rigid-jointed node a .
(a) Deformed configuration (dashed line) is after the snap through (with DR method).
(EA=1.855x10^6 lb, EI=9.27x10^3 lb/in^2)

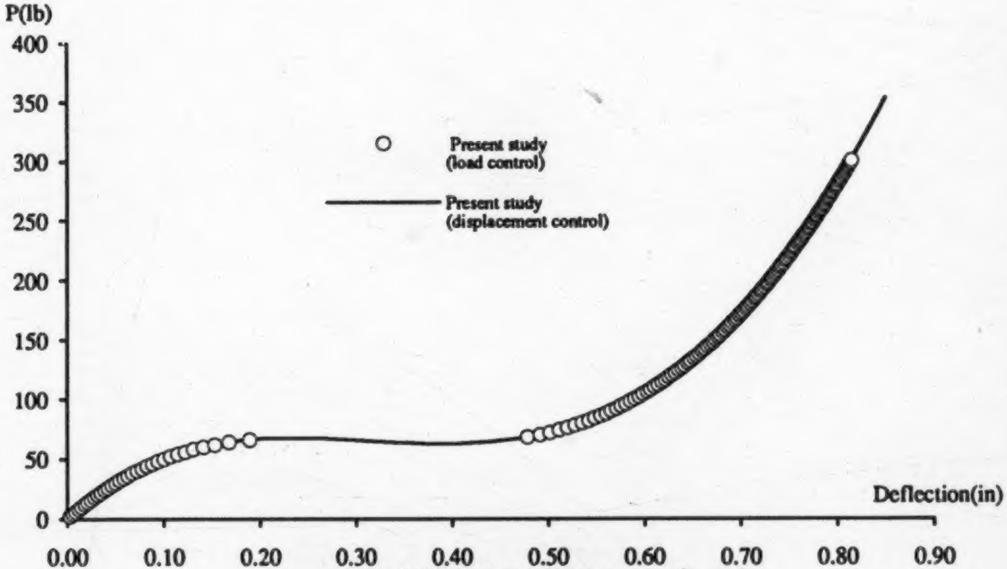


Figure (4.12b) Load-deflection curve for node a.

($E=10^7$psi, $G=3.33\text{x}10^6$ psi)

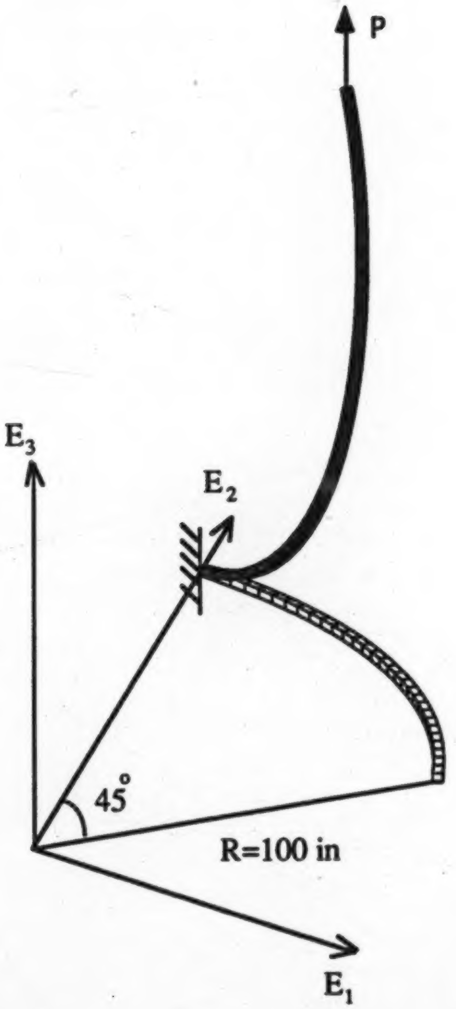


Figure (4-13): The deformed (with NR method) and undeformed configuration of initially curved cantilever beam with cross section of 1x1 in.The applied tip load P is out of the plane of initial curvature. (P=1250 lb)

	Inextensible case (Raboud *et al.* [29])			Extensible case		
k	E1 (in.)	E2 (in.)	E3 (in.)	E1 (in.)	E2 (in.)	E3 (in.)
1	69.1362	71.6529	15.2722	69.0609	71.7090	15.6527
2	65.4006	73.8728	27.5836	65.1787	74.0355	28.1189
3	61.0592	76.4225	36.4615	60.7362	76.6619	36.9854
4	56.9396	78.8072	42.7314	56.5712	79.0845	43.1851
5	53.2915	80.8861	47.2451	52.9133	81.1753	47.6212
6	50.1288	82.6567	50.9550	49.7592	82.9455	50.9027
7	47.3945	84.1679	53.1595	47.0418	84.4423	53.4113
8	45.0206	85.4561	55.1786	44.6873	85.7147	55.3864
9	42.9451	86.5639	56.8080	42.6307	86.8047	56.9816
10	41.1167	87.5240	58.1511	40.8193	87.7464	58.2983
11	39.4935	88.3625	59.2787	39.2108	88.5669	59.4057
12	38.0422	89.1002	60.2404	37.7719	89.2875	60.3519
13	36.7359	89.7536	61.0719	36.4760	89.9250	61.1716
14	35.5530	90.3359	61.7994	35.3020	90.4928	61.8900
15	34.4758	90.8580	62.4427	34.2326	91.0017	62.5261

Table (4.1): Geometry at the End of a Curved Cantilever Under the Action of a Dead Tip Load

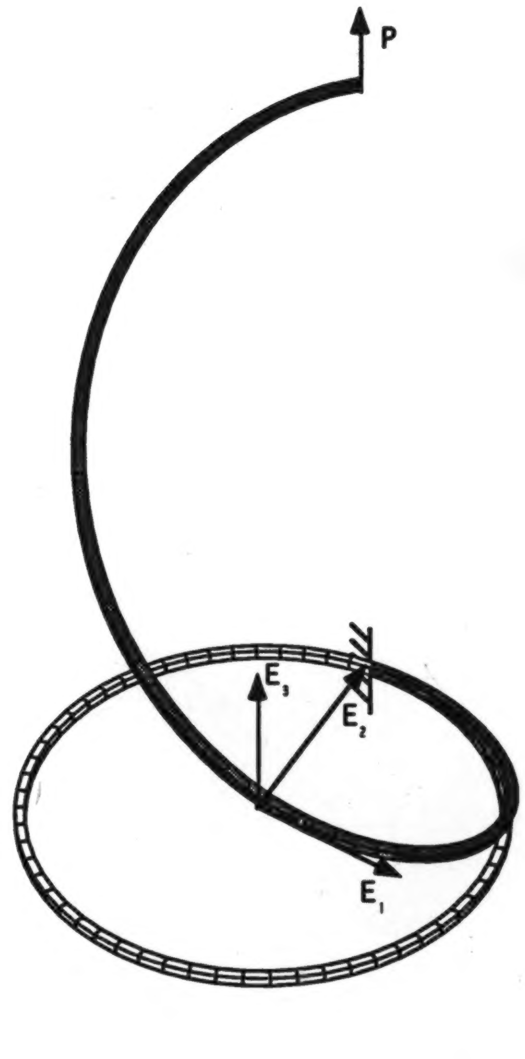


Figure (4-14): The deformed (with NR method) and undeformed configuration of initially curved cantilever beam with cross section of 1X1 in. The radius of circle in initial configuration is100 in. ($E=10^7$ psi, $G=3.33\times10^6$ psi)
The applied tip load P is out of the plane of initial curvature (P=32 lb).

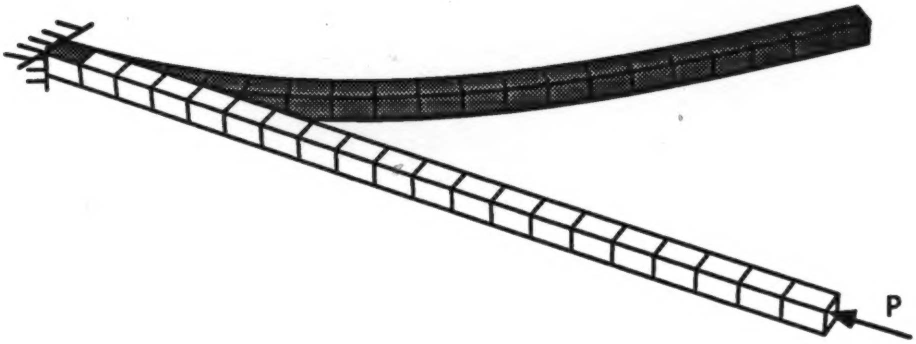
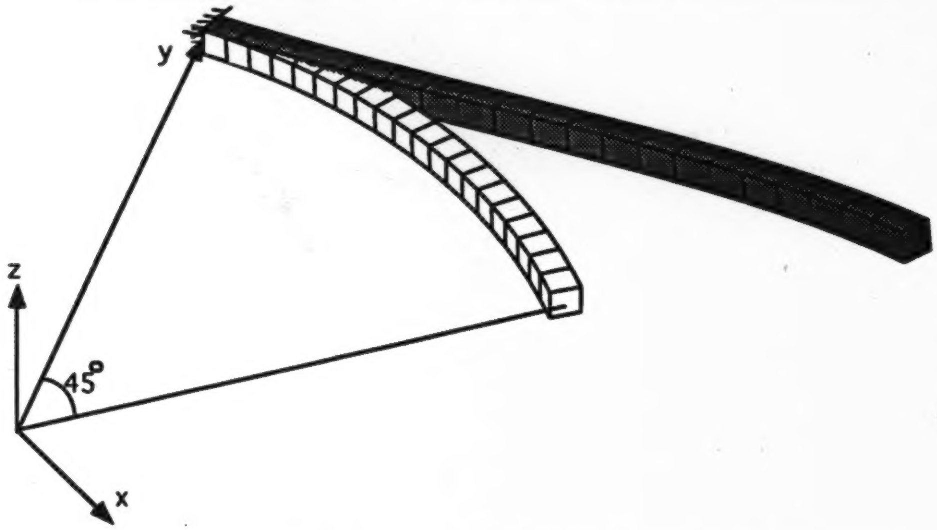
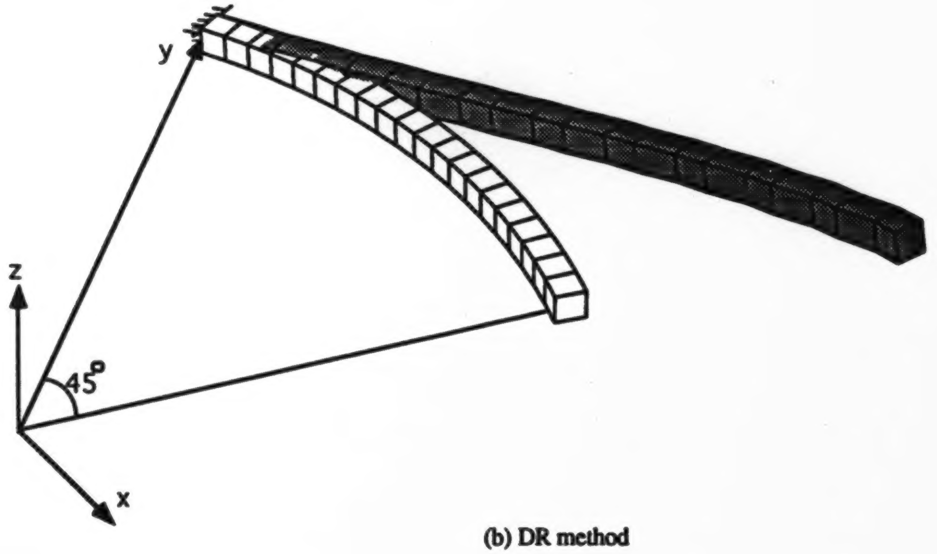


Figure (4-15): The buckled (with NR method) and initial configuration of a cantilever beam .(E=20000 psi, A=42 in^2, I=6482 in^4, L=400 in, P=2050 lb)



Example (4-16): The comparison of NR and DR methods for deformation of circular curved cantilever beam. (EI=8.33x10^5 lb.in^2 , GJ=5.5x10^5 lb.in^2 , EA=100 lb, radius=100 in) The applied tip loads are (P_x=P_y=P_z=100 lb, M_x=M_y=M_z=100 lb.in).

(a) NR method



(b) DR method

Node	Initial Cofiguration Coordinate		
	x (in.)	y (in.)	z (in.)
1	0	100.0000	0.0000
2	7.848	99.6900	0.0000
3	15.647	98.7685	0.0000
4	23.35	97.2362	0.0000
5	30.9	95.1043	0.0000
6	38.278	92.3860	0.0000
7	45.41	89.0978	0.0000
8	52.26	85.2602	0.0000
9	58.79	80.8967	0.0000
10	64.96	76.0344	0.0000
11	70.73	70.7031	0.0000

(a) The initial coordinate for different nodes

Node	Deformed Coordinate with NR			Deformed Coordinate with DR		
	x (in.)	y (in.)	z (in.)	x (in.)	y (in.)	z (in.)
1	0.0000	100.0000	0.0000	0.0000	100.0000	0.0000
2	11.8260	99.8952	0.2532	11.5810	99.8991	0.2550
3	23.6926	99.5348	0.9862	23.2039	99.5480	0.9839
4	35.5331	98.8561	2.1564	34.8002	98.8826	2.1344
5	47.2789	97.7969	3.7133	46.3020	97.8408	3.6507
6	58.8579	96.2975	5.5994	57.6392	96.3643	5.4745
7	70.1929	94.3033	7.7511	68.7375	94.4006	7.5453
8	81.2006	91.7685	10.0992	79.5168	91.9064	9.8008
9	91.7901	88.6600	12.5704	89.8897	88.8506	12.1776
10	101.8636	84.9617	15.0880	99.7620	85.2188	14.6111
11	111.3170	80.6800	17.5760	109.0339	81.0169	17.0367

(b) The numerical results for deformed configurations with NR and DR

Table (4.2): The comparison for deformed configurations between NR and DR

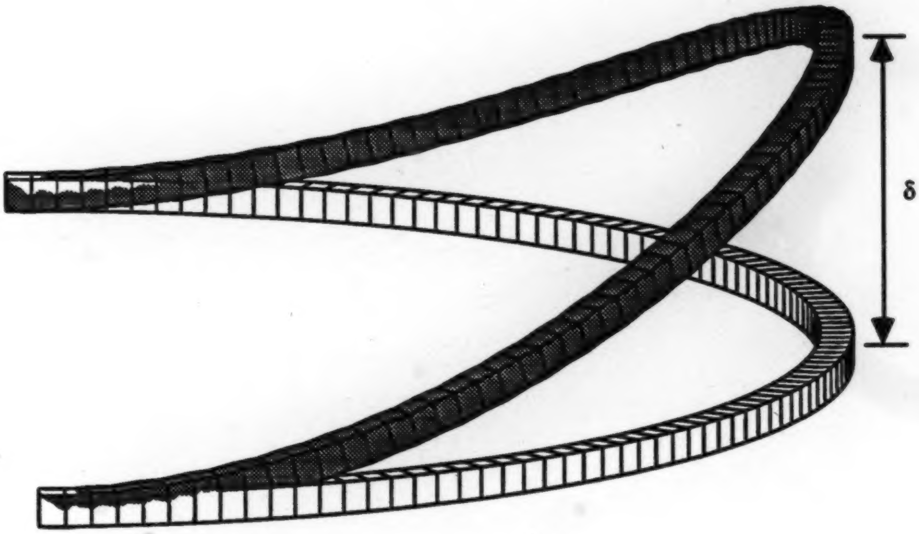


Figure (4-17): The undeformed and deformed configurations of fixed fixed semicircle under prescribed displacement and angle in the middle.
(using DR method)
($EI=8.33\times10^5$ lb.in^2, $GJ=5.5\times10^5$ lb.in^2, $\delta=34$ in, $EA=100$ lb, radius=100 in)

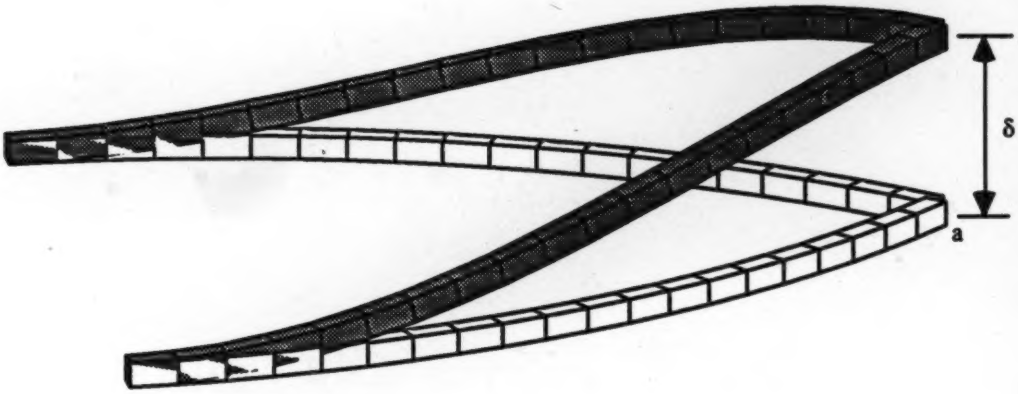


Figure (4-18): The undeformed and deformed configurations of a structure with two circular arcs (two 1/8 of a circle) under prescribed displacement and angle at node a. (using DR method)

(EI=8.33×10^5 lb.in^2 , GJ=5.5×10^5 lb.in^2 , EA=100 lb, δ=4.7 in, radius of arc=100 in)

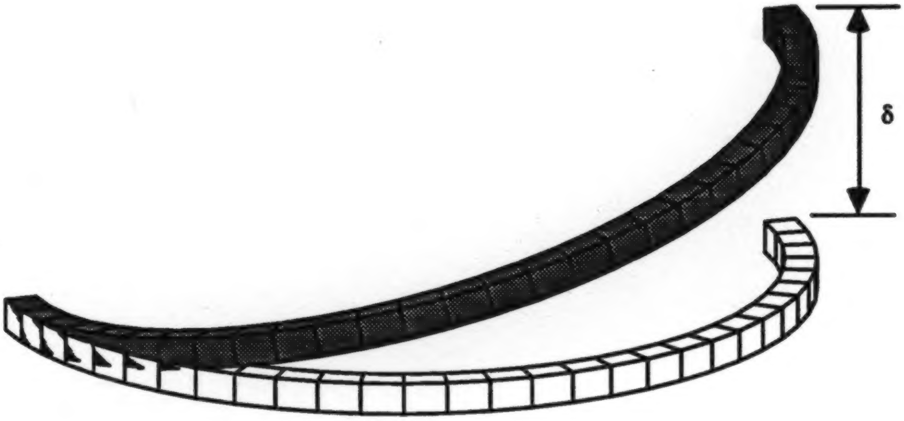


Figure (4-19): The undeformed and deformed configurations of fixed free semicircle under prescribed displacement and angle at the end. (using DR method) ($EI=8.33\times10^5$ lb.in^2, $GJ=5.5\times10^5$ lb.in^2, EA=100 lb, δ=56.5 in, radius=100 in)

Chapter 5

Inelastic Post-Buckling Analysis of Truss Structures by DR

5.1 Introduction

In the previous chapters the geometric nonlinearities were considered, assuming a linear elastic stress-strain relationship for the rod member. In reality however, nonlinear materials showing plastic behaviour could be involved. If we consider a one rod structure with such a material, increasing the load to the post-critical range, is called the failure of the structure. However, in space structures consisting of a large number of rods failure of an individual element does not necessarily result in the collapse of the structure. Instead the forces among the members in the neighborhood of the element could be distributed and the structure may still be capable of carrying increased loads, while the members which have failed will behave according to their post-critical characteristics. In fact considering of both types of nonlinearities becomes necessary when structural responses to exceptional loads are studied in order to assess ultimate strength or serviceability.

The aim of this chapter is to determine the complete load-deflection curve of trusses, far beyond the occurrence of the yield point, by considering material as well as geometric nonlinearities. The solution of the highly nonlinear governing equilibrium equations has been obtained by the dynamic relaxation method. Most investigators ([8], [14], [31], [33]) have used some scheme based on the well-known Newton-Raphson procedure where the condition that the determinant of the Jacobian matrix be nonzero is a basic requirement for the successful working of the method. For snap-through problems, however, the matrix becomes singular at the limit point. Therefore, the method needs special treatment to overcome this point without the failure of the numerical procedure. Negative elements on the leading diagonal of the stiffness matrix that may appear in the post-buckling range may pose additional problems when a Newton-Raphson type of method is used. With dynamic relaxation, however, all these problems are very easily surmounted. The difficulties in the vicinity of the limit points are over come by a straightforward implementation of a displacement incremental technique, while no provision is required for handling negative diagonal stiffness elements.

The inclusion of both geometric and material nonlinearities in the analysis of elasto-plastic trusses will be considered here. The solution of the highly nonlinear governing equilibrium equations has been obtained by the dynamic relaxation method in each increment of applied load and/or prescribed displacement.

The scope of this chapter is truss structures in which the axial force, F_1, is the only load to be calculated from the constitute law. To generalize to deformations with bending, we would need constitutive equations for moment versus curvature. Also when shear deformation is involved shear forces F_2 and F_3 can be calculated from the constitutive relationships (see Antman [1] for example). When shear is not considered, F_2 and F_3 are not given by any constitutive relationship but are rather determined by a balance of momentum as in the usual linear theory of elasticity.

5.2 Failure of conventional DR for inelastic material and introducing incremental DR

In this section, we present a counter example that shows why the standard DR doesn't work for inelastic materials and then explain the incremental-DR that we used to study inelastic materials. Let us consider a single rod under uniaxial load F as shown in Figure (5.1)(a). The rod has an initial length $L_0 = 1m$ and a cross section A =100 mm^2. The material is elastic hardening with the modulus of elasticity E = 2x10^8 KN/m^2, yield stress σ_y = 2x10^5 KN/m^2, and also e_1 = 4x10^7 KN/m^2. The applied force F is 10 N. Since $\sigma = 10^5 KN/m^2$ is less than σ_y, the rod is in elastic mode and the displacement of the rod at the loaded end is $X_eq = \sigma L_0/E$ which becomes 5x$10^{-4}m$ to maintain equilibrium. Now using DR, depending on different initial guesses we get different solutions which all satisfy the equilibrium equations as illustrated in Figure (5.1)(b). Incremental-DR is a method which should be used for structures subjected to both material and geometrical nonlinearities with path dependent behaviour. In this technique, loading and prescribed displacements are considered in an incremental way. We start with the structure in equilibrium and incrementing the load/prescribed displacement, the structure is equilibrated by solving the nonlinear equations of equilibrium using DR at the beginning and end of each increment. For each increment the configuration at the end of the previous increment will be considered as the initial guess for the next increment.

Inside each increment using DR the behavior of material considered on the specific path on the stress-strain curve and the irreversibility of inelastic strains has been taken into account only in passing from one increment of displacement stage to the subsequent one.

In each iteration of DR, the stress state should be compared with that of the beginning of the increment to determine whether the structure is being loaded or unloaded.

Considering Figure (5.1)(c) to start from initial configuration, initial condition is the beginning of increment. The DR in this increment will be forced to move on path $C'B'ABC$ depending on loading or unloading comparing to the beginning of the increment.

Considering Figure (5.1)(c) the beginning of the increment is at point A which will be the initial guess as well. The DR in this increment will be forced to move on path $B'AB$ depending on loading or unloading compared to the beginning of the increment. To get the right solution, the increment for applied load or applied displacement should be small, especially in the zones where the tangent modules has sudden and discontinuous changes.

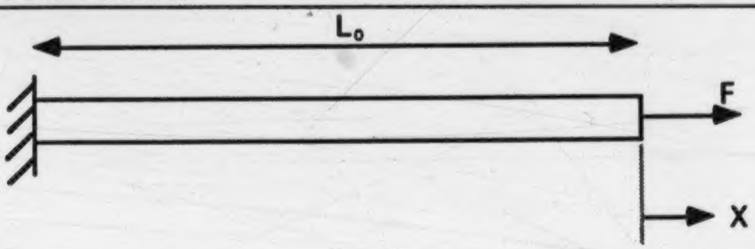
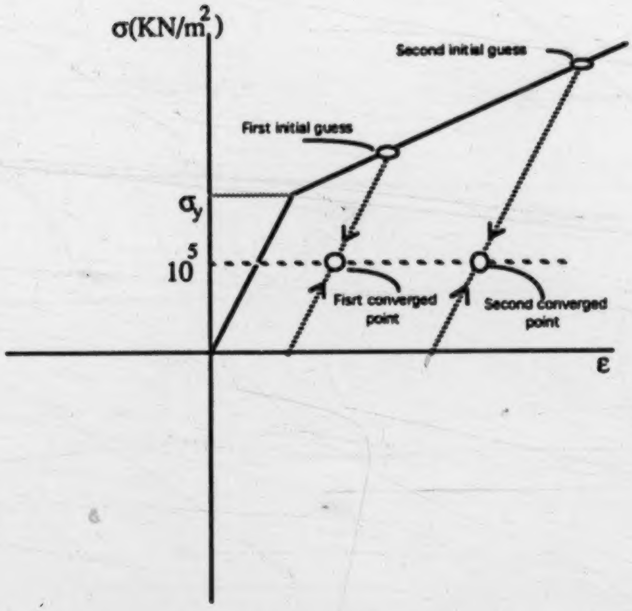
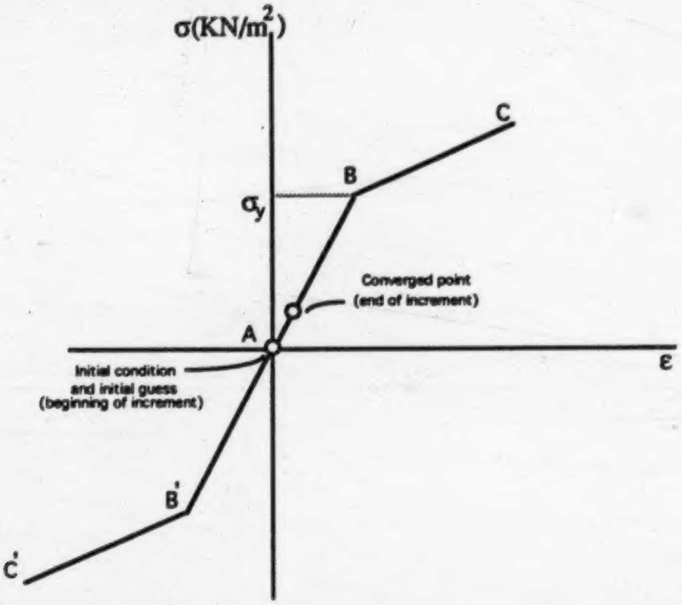


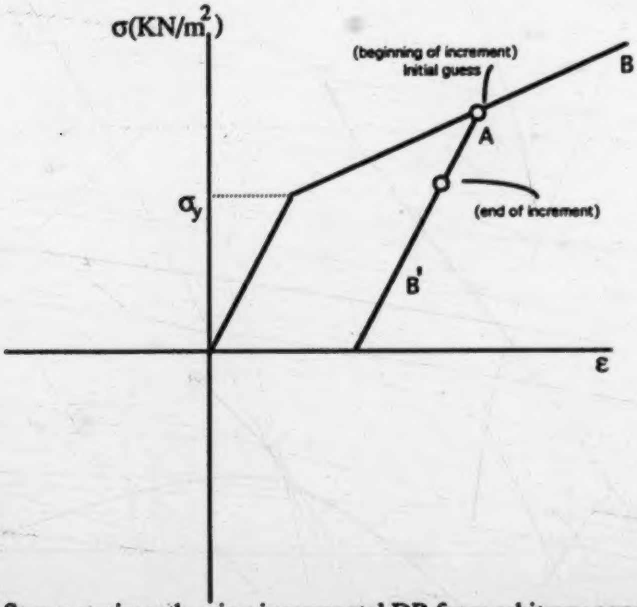
Figure (5.1) Figures used to justify using incremental-DR instead of conventional DR
(a): A single rod under uniaxial tension



(b): Stress-strain curve for using conventional DR


σ(KN/m²)
C
B
σy
Converged point
(end of increment)
A
Initial condition
and initial guess
(beginning of increment)
ε
B'
C'

(c): Stress-strain path using incremental DR from initial configuration


σ(KN/m²)
(beginning of increment)
Initial guess
B
A
σy
(end of increment)
B'
ε

(d): Stress-strain path using incremental DR from arbitrary condition

5.3 Formulation of the Problem

In the truss structures for all the rods M = 0, and therefore $\kappa = 0$. Considering Eqns. (2.14) a solution would be $\theta_1' = \theta_2' = \theta_3' = 0$. Since prime means derivative respect to reference arclength, so always θ_1, θ_2 and θ_3 would be constant along each rod and each rod will remain straight all the time. So we could take θ_1 to have any value (zero for convenience) and calculate θ_2, θ_3 from

$$\theta_2 = \sin^{-1}\frac{-r_3'}{\lambda}, \qquad \theta_3 = \tan^{-1}\frac{r_2'}{r_1'} \tag{5.1}$$

Also F_2 and F_3 are zero and $F_1 = \sigma A$ where σ is the axial stress in the rod and A is its cross section area. So Eqns. (4.24) can be simplified as $\eta_1 = F_1', \eta_2 = 0$ and $\eta_3 = 0$, then Eqn. (4.25) becomes.

$$T_{1j} F_1' \mathbf{E}_j = \rho A \ddot{r}_j \mathbf{E}_j + cA\dot{r}_j \mathbf{E}_j \qquad \text{where} \qquad j = 1, 2, 3 \tag{5.2}$$

5.3.1 Constitutive Laws

For the materials showing inelastic behaviour the axial stress value is not uniquely defined, for a given strain value. So the history of loading is required to yield a value of stress corresponding to the given load/displacement path. Figure (5.2) shows the stress-strain relations of the four different materials that we will consider here. For such materials all the possible paths on the stress-strain curve should be carefully coded. The stress-strain coding for the different paths for Elastic-hardening and nonlinear-elastic materials is given in Figure (5.3). Figures (5.4) (a) and (b) show the constitutive law flow charts for Elastic-hardening and nonlinear-elastic materials respectively. The following is the list of the variables used in coding the constitutive law for different materials:

- $\epsilon_{yt}, \sigma_{yt}$ are stress and strain for the yield point in tension
- $\epsilon_{yc}, \sigma_{yc}$ are stress and strain for the yield point in compression

- ϵ_o, σ_o are stress and strain in previous step

- ϵ_n, σ_n are stress and strain in present step

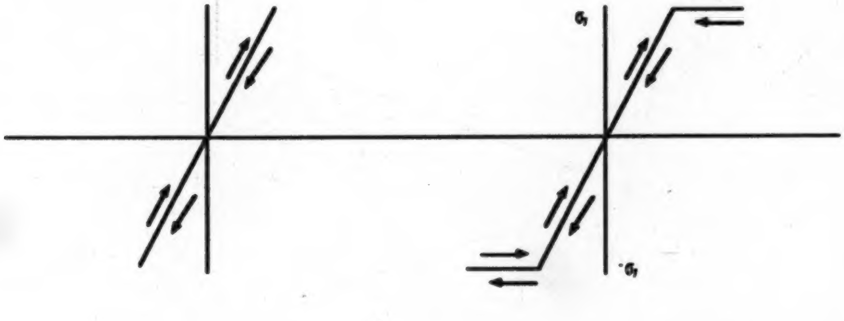
- ϵ_u for each step is the strain for the point of intersection of the lines with code 0 and code 1 shown in Figure (5.3(a) for Elastic-hardening material.

$$\epsilon_u = \frac{\sigma_{yt} - \sigma_o + E\epsilon_o - e_1\epsilon_{yt}}{E - e_1} \tag{5.3}$$

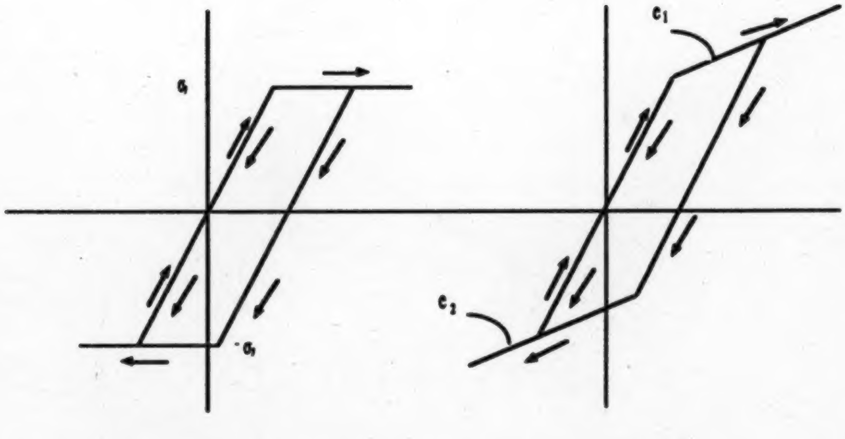
- ϵ_d for each step is the strain for the point of intersection of the lines with code 0 and code 2 shown in Figure (5.3(a) for Elastic-hardening material.

$$\epsilon_d = \frac{\sigma_{yc} - \sigma_o + E\epsilon_o - e_2\epsilon_{yc}}{E - e_2} \tag{5.4}$$

It is clear that setting $e_1 = e_2 = 0$ in the Elastic-hardening case yields the Elastic-perfectly plastic behaviour.



(a) Linear-Elastic (b) Nonlinear-Elastic



(c) Elastic-perfectly plastic (d) Elastic-hardening

Figure (5.2): Stress-strain relation for different materials

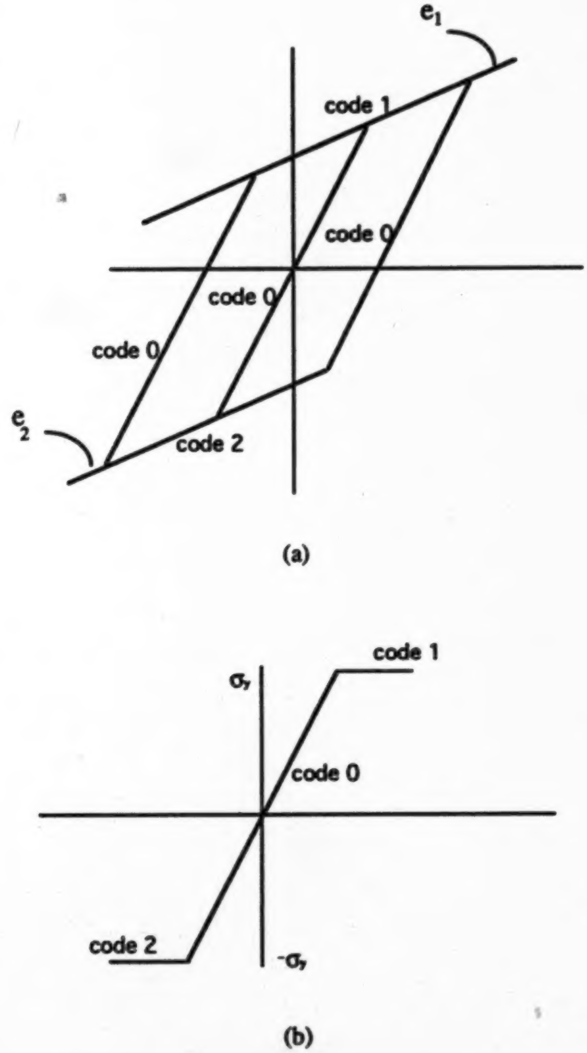

e_1
code 1
code 0
code 0
code 0
e_2
code 2
(a)
code 1
σ_y
code 0
code 2
$-\sigma_y$
(b)

Figure (5.3) Coding for different paths
(a) Elastic-hardening
(b) Nonlinear-Elastic

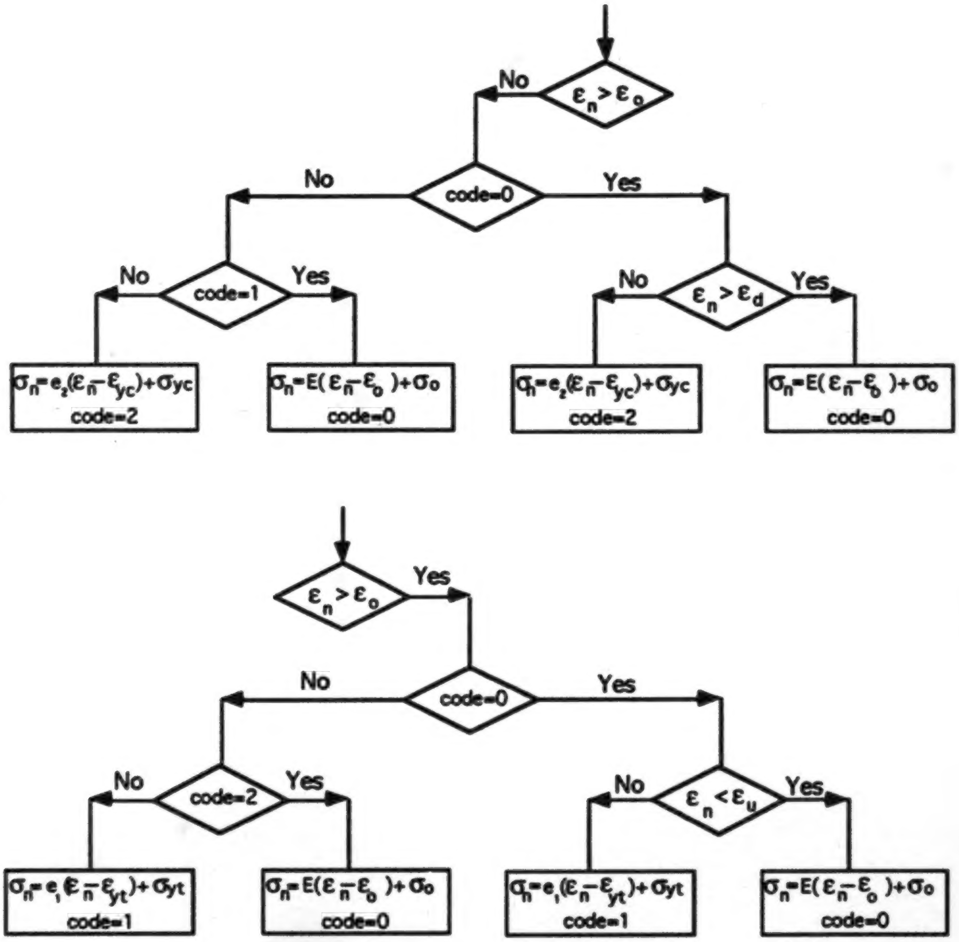
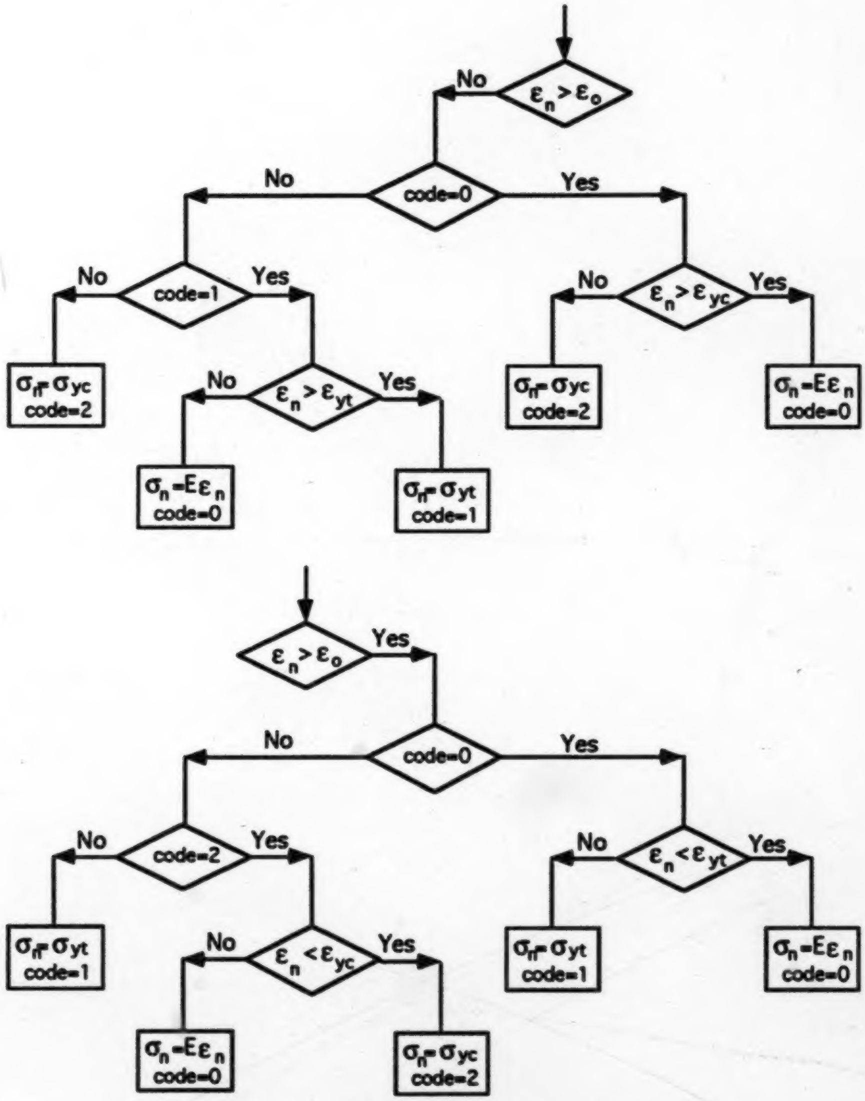

$\varepsilon_n > \varepsilon_o$
No
code=0
No
Yes
code=1
No
Yes
$\varepsilon_n > \varepsilon_d$
No
Yes
$\sigma_n = e_2(\varepsilon_n - \varepsilon_{yc}) + \sigma_{yc}$ code=2
$\sigma_n = E(\varepsilon_n - \varepsilon_o) + \sigma_o$ code=0
$\sigma_n = e_2(\varepsilon_n - \varepsilon_{yc}) + \sigma_{yc}$ code=2
$\sigma_n = E(\varepsilon_n - \varepsilon_o) + \sigma_o$ code=0
$\varepsilon_n > \varepsilon_o$
Yes
code=0
No
Yes
code=2
No
Yes
$\varepsilon_n < \varepsilon_u$
No
Yes
$\sigma_n = e_1(\varepsilon_n - \varepsilon_{yt}) + \sigma_{yt}$ code=1
$\sigma_n = E(\varepsilon_n - \varepsilon_o) + \sigma_o$ code=0
$\sigma_n = e_1(\varepsilon_n - \varepsilon_{yt}) + \sigma_{yt}$ code=1
$\sigma_n = E(\varepsilon_n - \varepsilon_o) + \sigma_o$ code=0

Figure (5.4) Flow chart for different materials
(a) For Elastic-hardening material


No
$\varepsilon_n > \varepsilon_o$
No
code=0
Yes
No
code=1
Yes
No
$\varepsilon_n > \varepsilon_{yc}$
Yes
$\sigma_n = \sigma_{yc}$ code=2
No
$\varepsilon_n > \varepsilon_{yt}$
Yes
$\sigma_n = \sigma_{yc}$ code=2
$\sigma_n = E\varepsilon_n$ code=0
$\sigma_n = E\varepsilon_n$ code=0
$\sigma_n = \sigma_{yt}$ code=1
$\varepsilon_n > \varepsilon_o$
Yes
No
code=0
Yes
No
code=2
Yes
No
$\varepsilon_n < \varepsilon_{yt}$
Yes
$\sigma_n = \sigma_{yt}$ code=1
No
$\varepsilon_n < \varepsilon_{yc}$
Yes
$\sigma_n = \sigma_{yt}$ code=1
$\sigma_n = E\varepsilon_n$ code=0
$\sigma_n = E\varepsilon_n$ code=0
$\sigma_n = \sigma_{yc}$ code=2

(b) Flow chart for the Nonlinear-Elastic material

5.4 DR Algorithm for Inelastic Materials

Since the behaviour of the structure for the inelastic material is path-dependent, to get the right solution, the time increment should be small, especially in the zones where the tangent modules has sudden and discontinous changes. The DR algorithm then can be used as the following steps

1. Choose $n = 0; r_1, r_2, r_3$ given; $\dot{r}_1 = 0$, $\dot{r}_2 = 0$, $\dot{r}_3 = 0$
2. Use $\lambda = |\mathbf{r}'|$, $\theta_2 = \arcsin \frac{-r'_3}{\lambda}$ and $\theta_3 = \arctan \frac{r'_2}{r'_1}$ to get λ, θ_2 and θ_3
3. Having $\epsilon_n = 1 - \lambda$, use proper algorithm to find σ_n
4. Use $F_1 = A\sigma_n$ to find F_1
5. Check R_1, R_2 and R_3 at the n^{th} step using $R_i^{(n)} = (T_{1i}F'_1)^{(n)}$ where i=1,2,3
   if $R_1^{(n)} \approx 0$, $R_2^{(n)} \approx 0$ and $R_3^{(n)} \approx 0$ stop, otherwise continue.
6. For $j = 1, 2, 3$; calculate

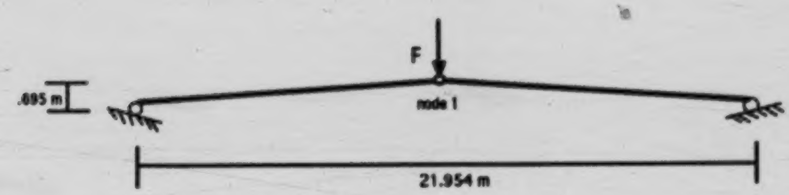
$$\begin{cases} \dot{r}_j^{(\frac{1}{2})} = \frac{\Delta t}{2m_{r_j}} R_{r_j}^{(0)} & \text{for } n = 0 \\ \dot{r}_j^{(n+\frac{1}{2})} = (\frac{1}{\Delta t} m_{r_j} + \frac{1}{2} c_{r_j})^{-1} [(\frac{1}{\Delta t} m_{r_j} - \frac{1}{2} c_{r_j}) \dot{r}_j^{(n-\frac{1}{2})} + R_{r_j}^{(n)}] & \text{for } n \neq 0 \\ {r_j}^{(n+1)} = {r_j}^{(n)} + \Delta t \dot{r}_j^{(n+\frac{1}{2})} & \text{for all } n \end{cases}$$

7. $n = n + 1$; go to 2.

5.5 Numerical Results

Two examples were examined and their response to different stress-strain relations of Figure (5.2) were compared. Papadrakakis [24] also solved these problems by DR method using the *beam-column* approach in the element formulation. Our results are in good agreement with his results.

Example (5-1). The two member truss is shown in Figure (5.5a). For each member of truss the modulus of elasticity is $E = 2.06\text{x}10^8\ KN/m^2$, the yield stress is $\sigma_y = 2.35\text{x}10^5\ KN/m^2$, and for Elastic-hardening material $e_1 = 4\text{x}10^7\ KN/m^2$, $e_2 = 8\text{x}10^7\ KN/m^2$. The comparison of stress-strain curves for each member of truss is shown in Figure (5.5b). The load-vertical displacement response for apex node (node 1 of Figure (5.5a)) for different stress-strain relations is shown in Figure (5.5c). The comparison of stress-vertical displacement response node 1 for different stress-strain relations is shown in Figure (5.5d). Wherever applicable, on all graphs, the small letters indicate the corresponding stress-strain relations from Figure (5.2), while capital letters correspond to the characteristic points of the stress-strain curve. On graphs (5.5b),(5.5c) and (5.5d), at point A curves corresponding to other materials deviate from Linear-Elastic case. Point B corresponds to the maximum compressive stress/strain. The strain at point B is the same for all the materials whereas the value of stress is different. This point corresponds to a deformed configuration in which rods are on the line connecting the two supports (vertical apex displacement is $0.695m$). No force would be required to keep equilibrium at this point. After this point to keep equilibrium satisfied we should change the direction of the applied force F. However, since the internal force is still compressive, the stress in each member of truss would stay compressive. At point C_b, Nonlinear Elastic and Linear Elastic curves join together once again. In Figures (5.5c) and (5.5d) points C_b and A are symmetric with respect to point B. Points A and C_b are corresponding to the deformed configurations where apex node would be at the same height above and below the deformed configuration corresponding to point B. As we continue to increase the force, after sometime we should change the direction of applied force F which corresponds to the intersection with the ϵ axes in Figure (5.5b), Z axes in Figures (5.5c) and (5.5d). For Linear-Elastic and Nonlinear-Elastic materials, this corresponds to the deformed configuration where node 1 in is at the same height as initial configuration, below the deformed configuration corresponding to point B. Whereas for Elastic-perfectly plastic and Elastic-hardening at this stage we would have compression strain. Points D_c, D_b correspond to the change in the slope of stress-strain curves. All curves pass through point E which is the yield point in tension and from that point on Nonlinear Elastic separate from Linear Elastic. If we unload from point E, all the materials behave the same way and curves move toward point C_b on the Linear Elastic curve. Whereas if we unload from before or after point E it would not be the case. Figure (5.5e) through (5.5j) shows the comparison of our results with that of Papadrakakis [24].



(a) The geometry of two member truss.

Figure (5.5): The two member truss with rectangular cross section of 2.54 x 2.54 m.
The modulus of elasticity is, $E=2.06 \times 10^8$ kN/m^2.
The yield stress is $\sigma_y = 2.35 \times 10^5$ kN/m^2.
For Elastic -hardening material $e_1=4 \times 10^7$ kN/m^2 and $e_2=8 \times 10^7$ kN/m^2.

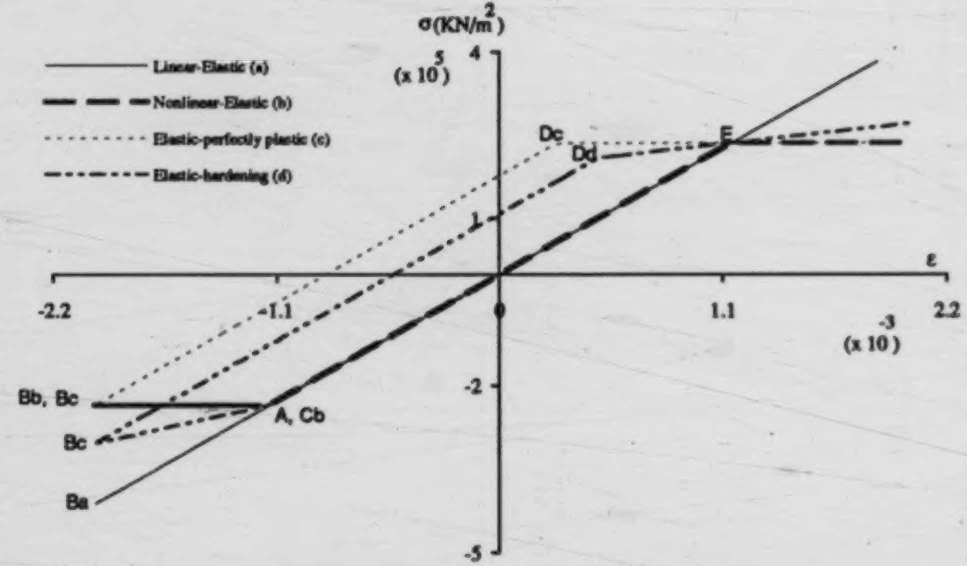


Figure (5.5b) Comparison of stress-strain curves for each member of truss.

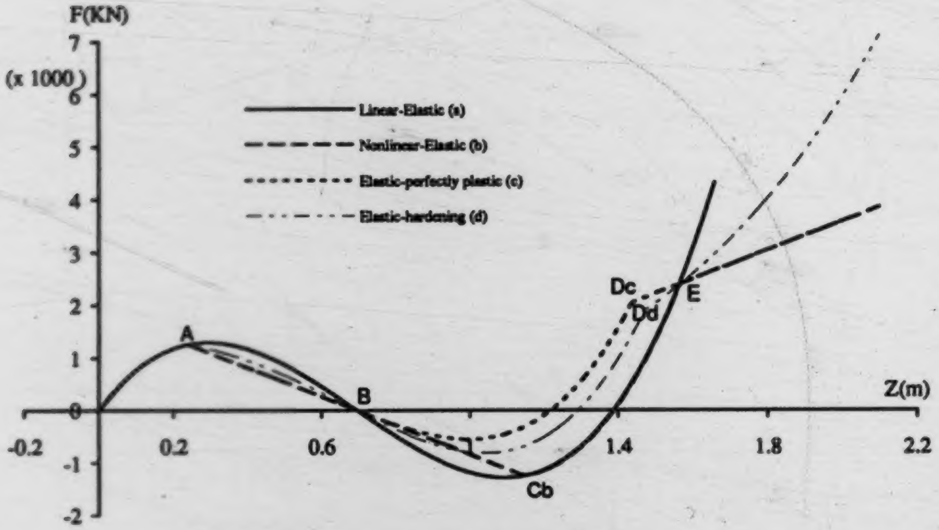


Figure (5.5c) Comparison of load-vertical displacement curves for node 1.

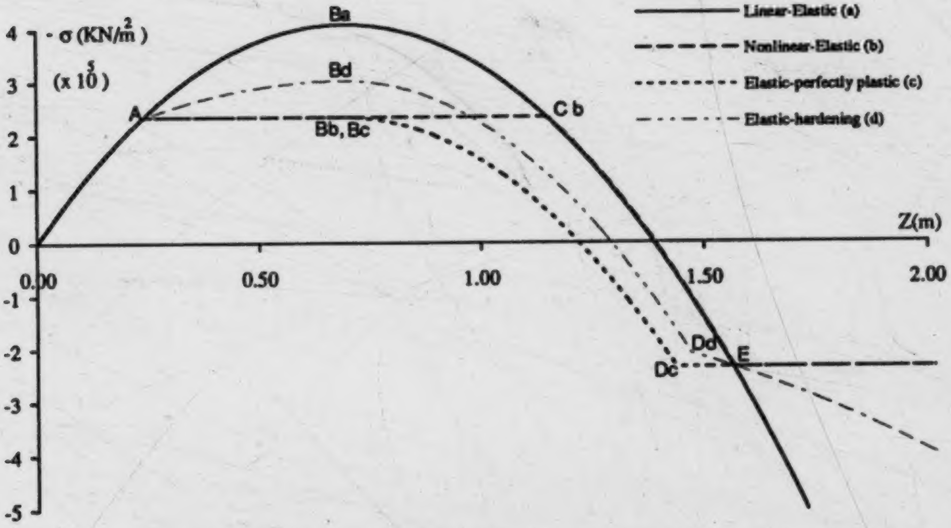


Figure (5.5d) Comparison of stress-vertical displacement curves for each member of truss.

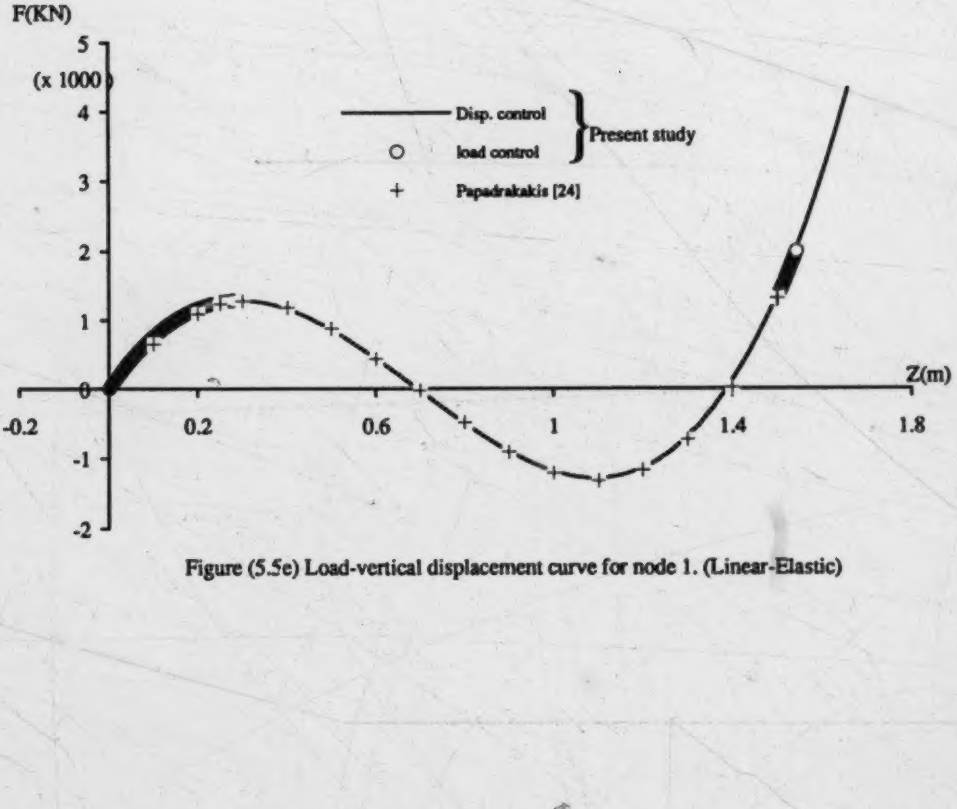


Figure (5.5e) Load-vertical displacement curve for node 1. (Linear-Elastic)

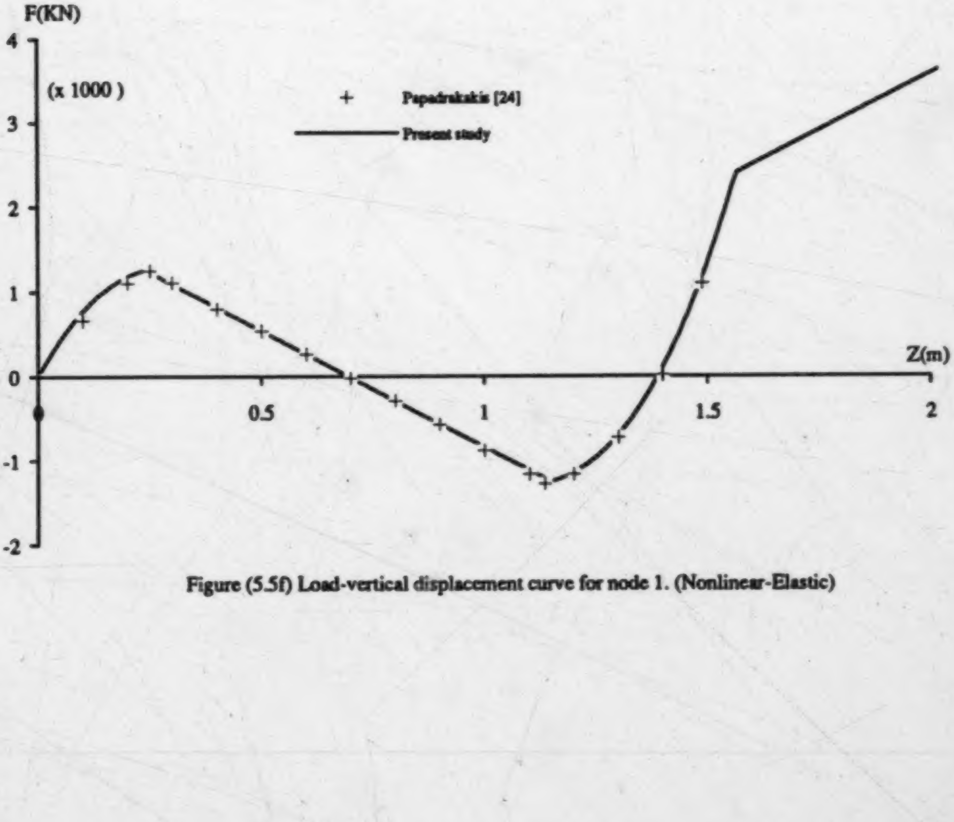


Figure (5.5f) Load-vertical displacement curve for node 1. (Nonlinear-Elastic)

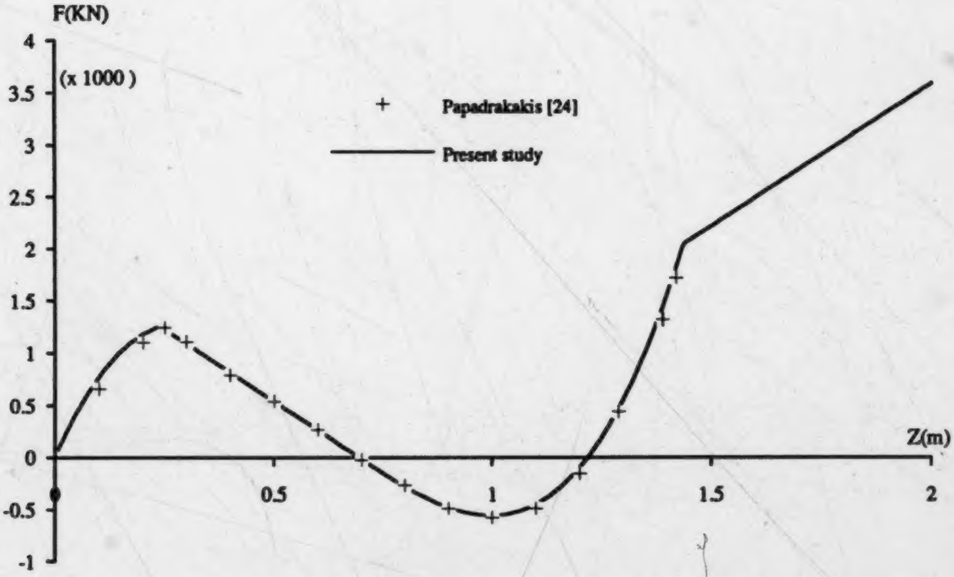


Figure (5.5g) Load-vertical displacement curve for node 1. (Elastic-perfectly plastic)

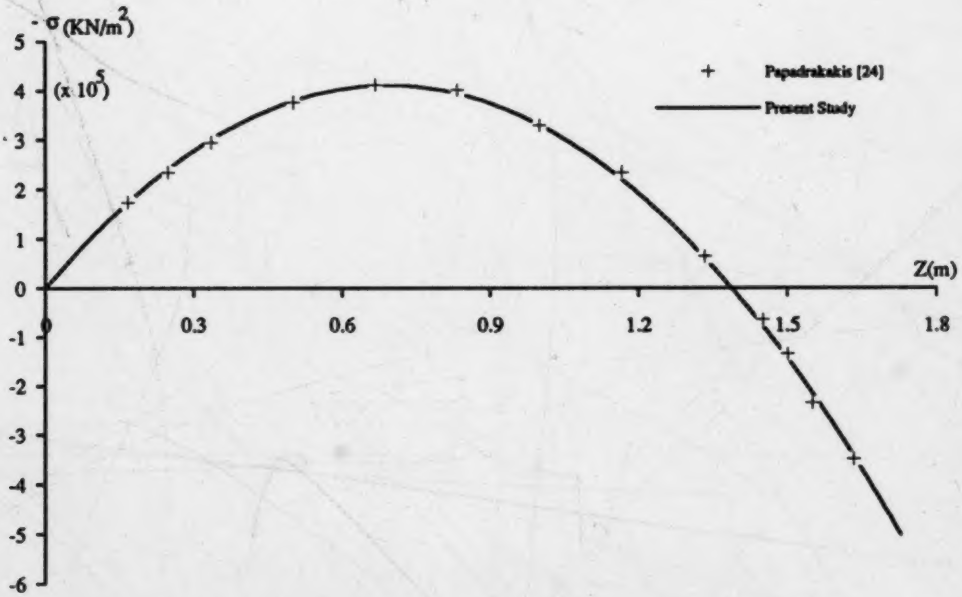


Figure (5.5h) Stress-vertical displacement curve for each member of truss. (Linear-Elastic)

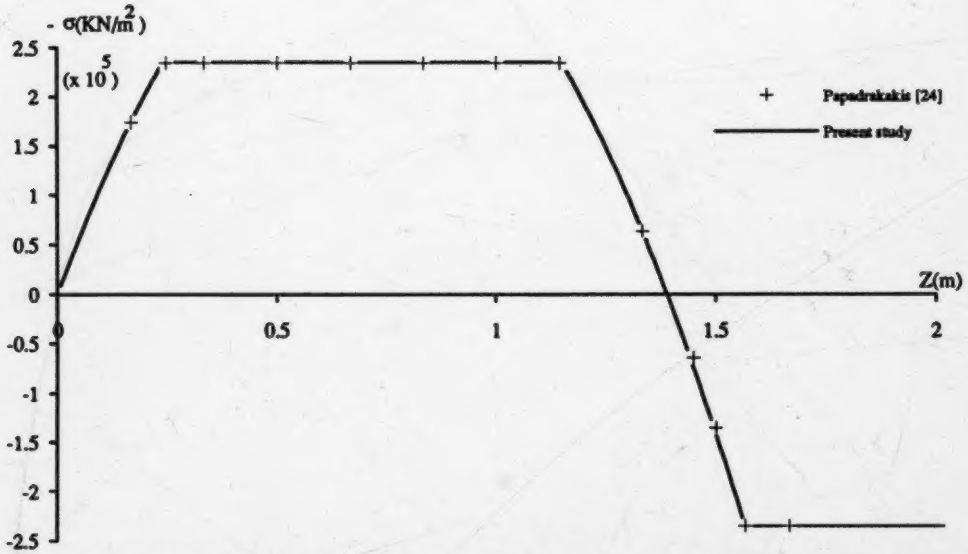


Figure (5.5i) Stress-vertical displacement curve for each member of truss. (Nonlinear-Elastic)

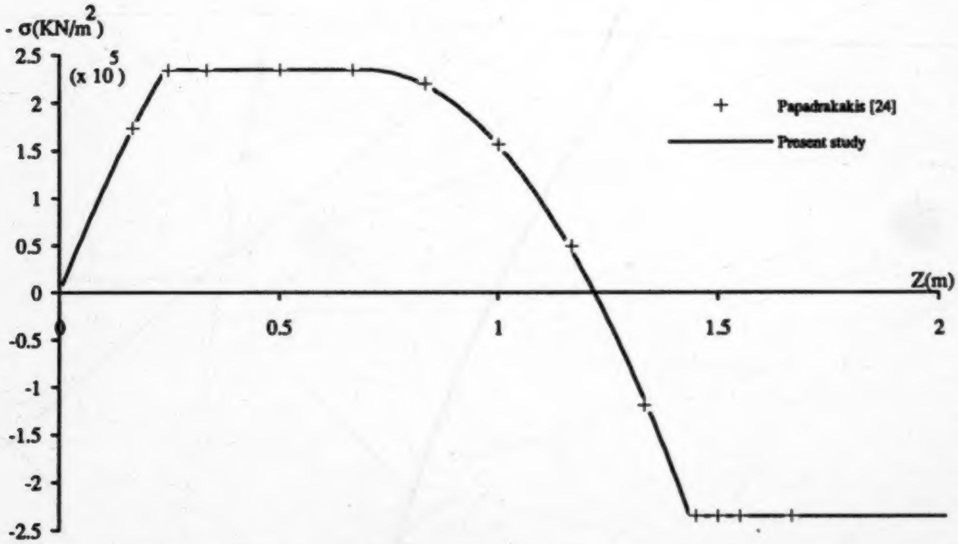
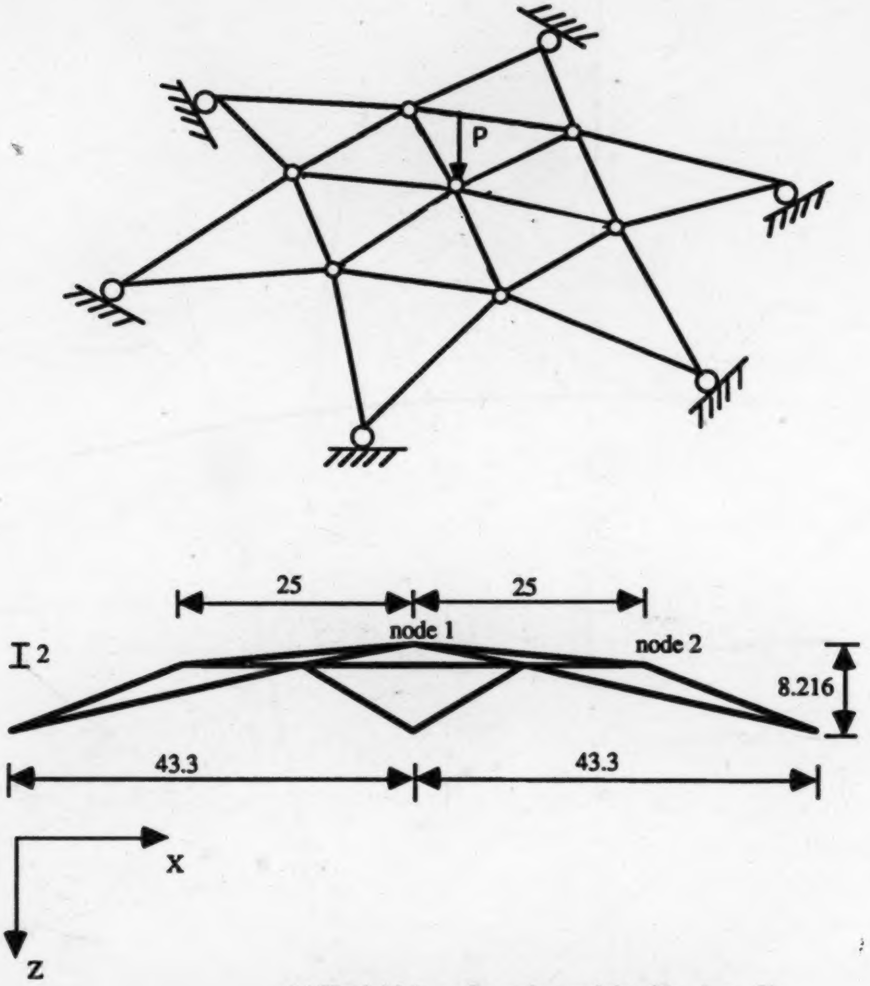


Figure (5.5j) Stress-vertical displacement curve for each member of truss. (Elastic-perfectly plastic)

Example (5-2). The 24-member shallow dome is shown in Figure (5.6a). For all members the modulus of elasticity is $E = 2.06\text{x}10^8\ KN/m^2$, the yield stress is $\sigma_y = 2.35\text{x}10^5\ KN/m^2$, and for Elastic-hardening material $e_1 = 4\text{x}10^7\ KN/m^2$, $e_2 = 8\text{x}10^7\ KN/m^2$. The comparison of stress-strain curves for member 1 of structure is shown in Figure (5.6b). The load-vertical displacement response for apex node (node 1) for different stress-strain relations is shown in Figure (5.6c). Again here, at point A curves corresponding to other materials deviate from Linear-Elastic case. Point B corresponds to the maximum compressive stress/strain, but the strain at point B is different for different materials. At point C_b, Nonlinear Elastic and Linear Elastic curves join together once again. All curves pass through point E which is yield point in tension. Figure (5.6d) shows the response of the displacement of node 2 in the X direction with respect to the apex load P. Figures (5.6e), (5.6f) and (5.6g) show the axial stress of members 1, 2 and 3 versus vertical apex displacement curves respectively, for different materials. Figures (5.6h), (5.6i), (5.6j) and (5.6k) compare the axial stress of members 1, 2 and 3 for different materials Linear-Elastic, Nonlinear-Elastic, Elastic-perfectly plastic and Elastic-hardening. From Figure (5.6d) it can be seen that curve for Nonlinear Elastic has a discontinuity at point B, where the load P changes sign, from point B_b to B'_b. Considering Figure (5.6i) shows that at this point members type 1 and type 2 are in the same plane in the deformed configuration and all of these members are at their limiting stress (members type 1 in compression and members type 2 in tension and members type 3 are stress-free). When node 1 is displaced in the Z direction further, a tensile stress is developed in members type 3 and consequently the stress in members type 1 will be less than limiting case to maintain equilibrium. This causes a jump in the strain for this specific material and hence a discontinuity is observed in the force-displacement curve of Figure (5.6d).



(a) The initial configuration and the side-view of it.

Figure (5-6): The space structure with ball-jointed nodes, the vertical force applied at central node 1. (all dimensions are in cm)
The cross section is rectangular of 2.54 x 2.54 cm.
The modulus of elasticity is, $E=2.06 \times 10^8$ kN/m^2.
The yield stress is $\sigma_y = 2.35 \times 10^5$ kN/m^2.
$e_1=4 \times 10^7$ and $e_2=8 \times 10^7$.

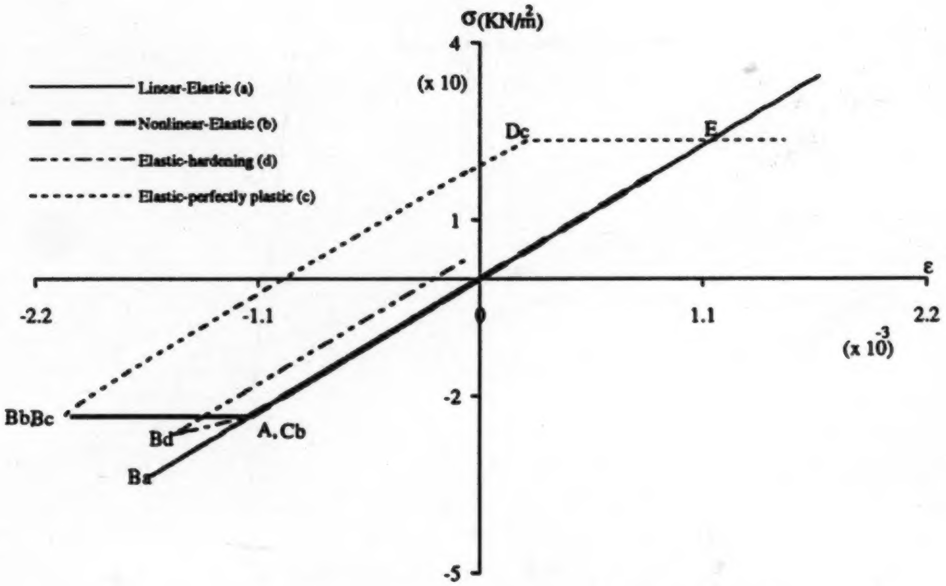


Figure (5.6b) Comparison of stress-strain curves for member 1

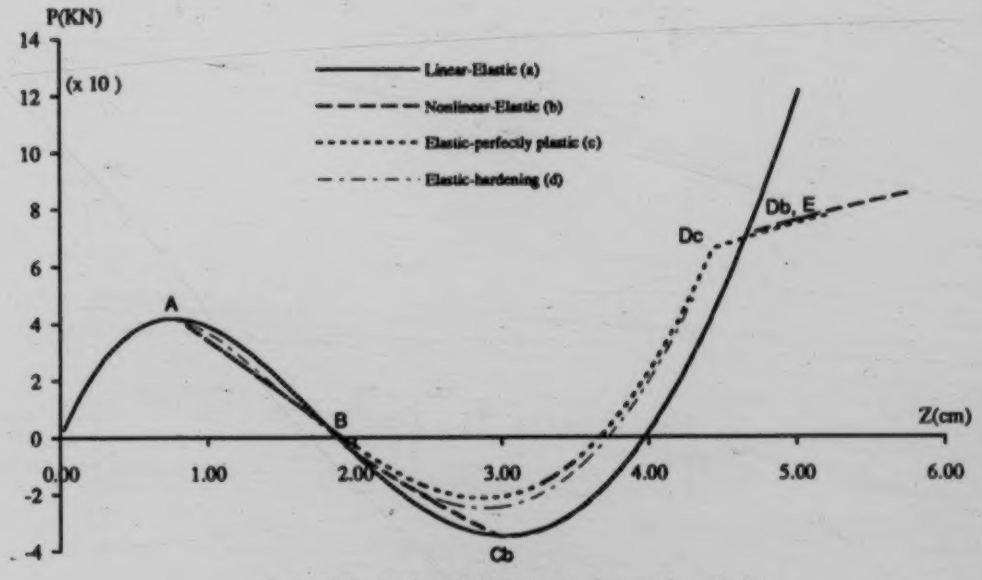


Figure (5.6c) Load-vertical displacement curves for node 1

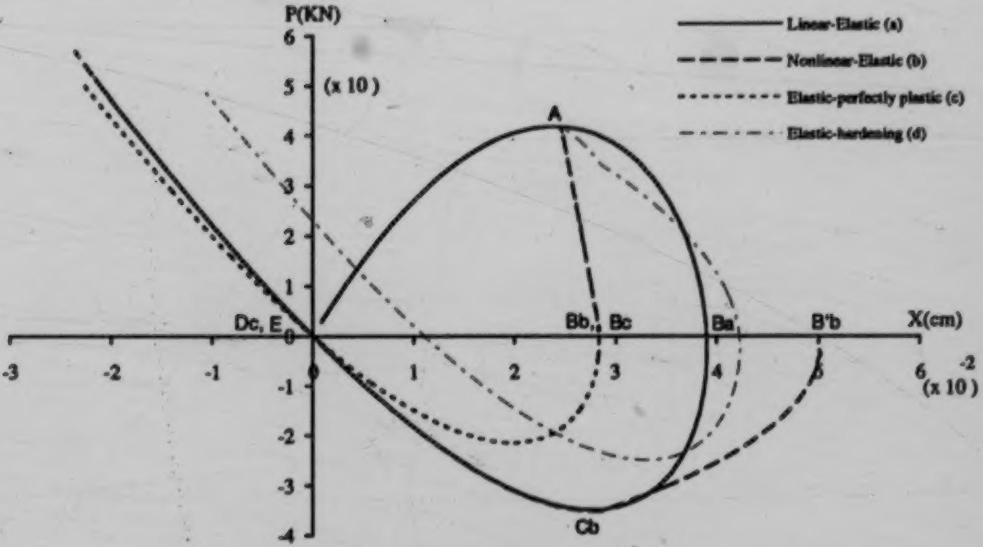


Figure (5.6d) Comparison of load-horizontal displacement curves for node 2

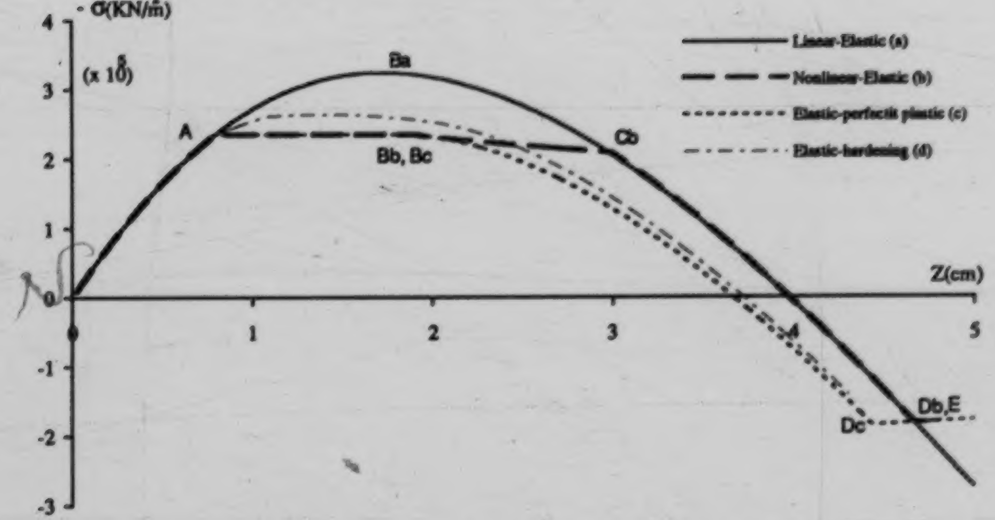


Figure (5.6e) Axial stress of member 1 vs. vertical apex displacement curves.

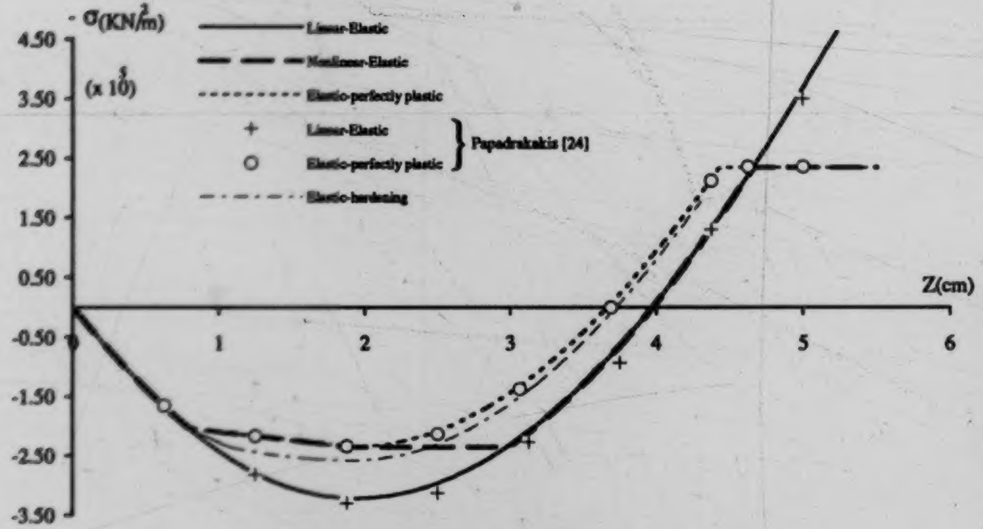


Figure (5.6f) Axial stress of member 2 vs. vertical apex displacement curves.

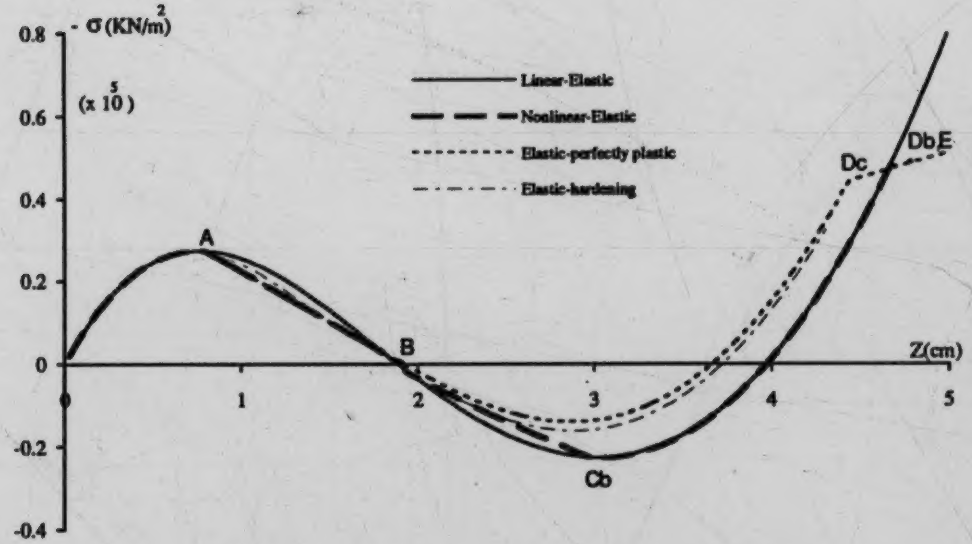


Figure (5.6g) Axial stress of member 3 vs. vertical apex displcement curves.

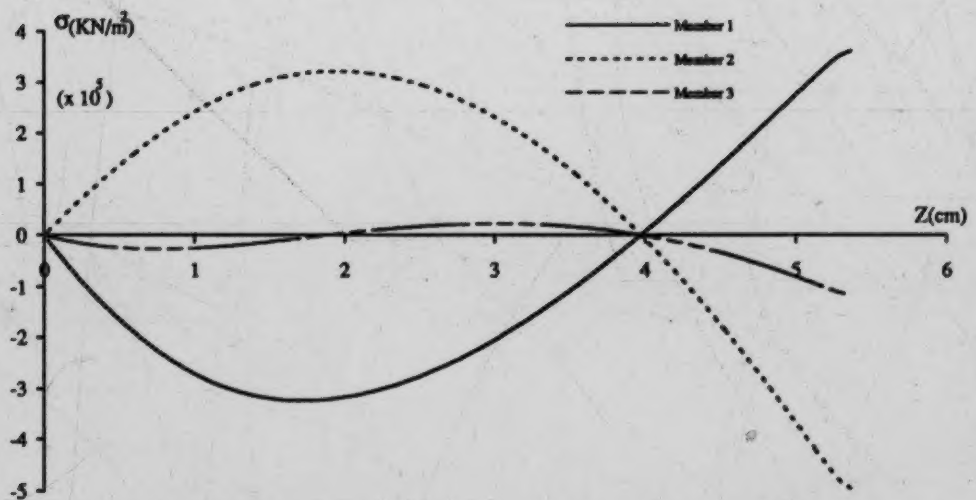


Figure (5.6h) Axial stress of members vs. vertical apex displacement curves. (Linear-Elastic)

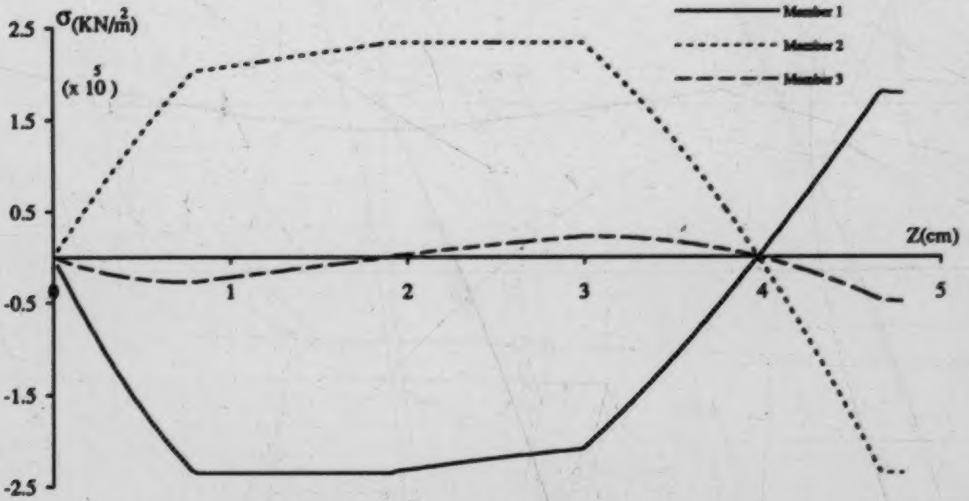


Figure (5.6i) Axial stress of members vs. vertical apex displacement curves. (Nonlinear-Elastic)

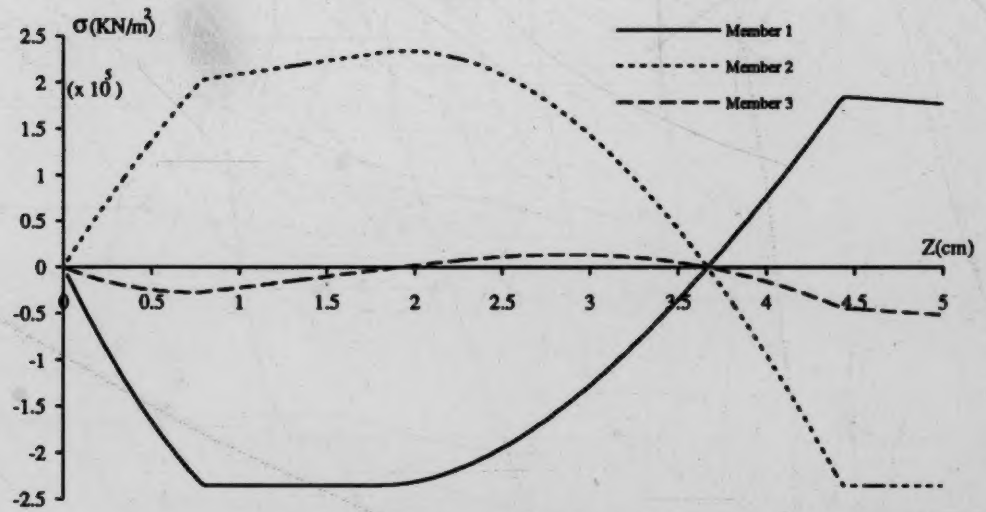


Figure (5.6j) Axial stress of members vs. vertical apex displacement curves. (Elastic-perfectly plast

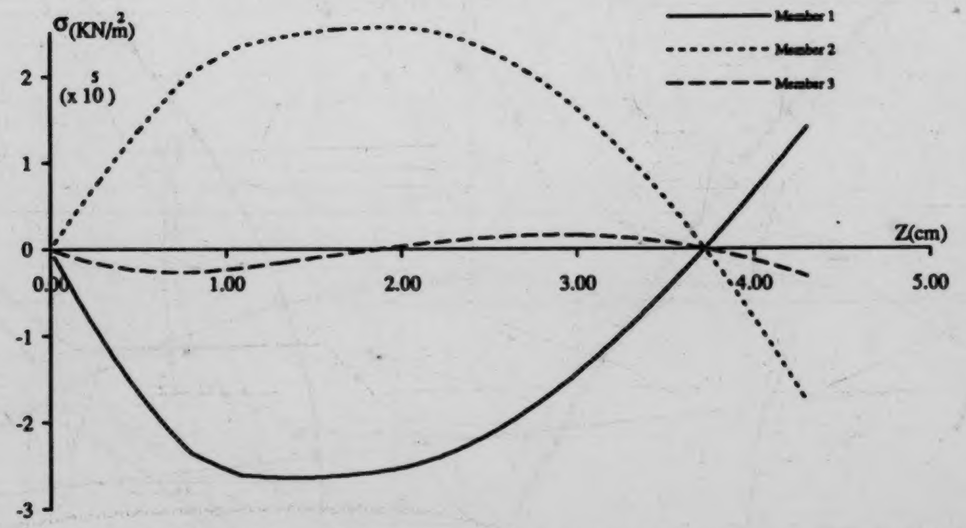


Figure (5.6k) Axial stress of members vs. vertical apex displacement curves. (Elastic-hardening)

The comparison of our results with the results of Papadrakakis [24] for load-vertical displacement of node 1 for Linear-Elastic, Nonlinear-Elastic and Elastic-perfectly plastic are given in Figures (5.6L), (5.6m) and (5.6n) respectively. Also the comparison of our results with the results of Papadrakakis [24] for load-horizontal displacement of node 2 for Linear-Elastic, Nonlinear-Elastic and Elastic-perfectly plastic are given in Figures (5.6o), (5.6p) and (5.6q) respectively. He did not derive any result for Elastic-hardening.

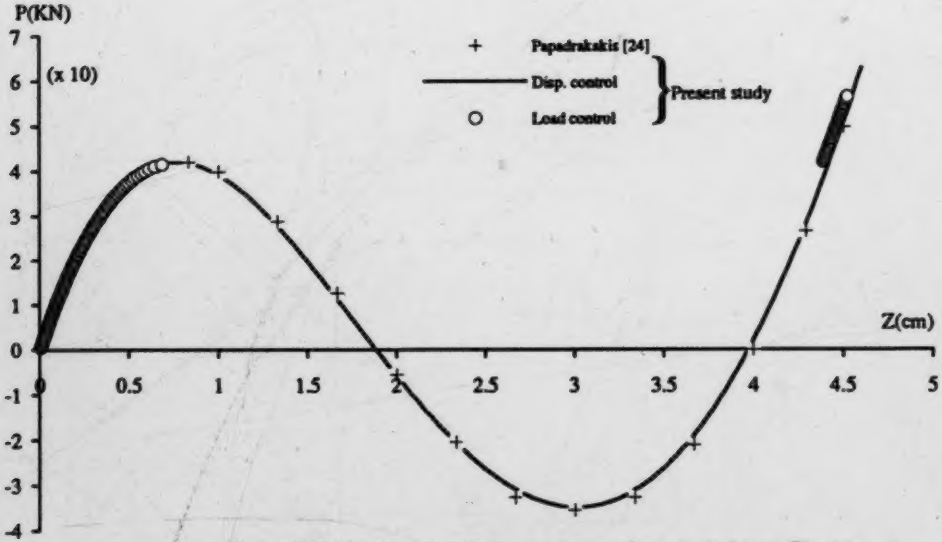


Figure (5.6L) Load-vertical displacement curve for node 1. (Linear-Elastic)

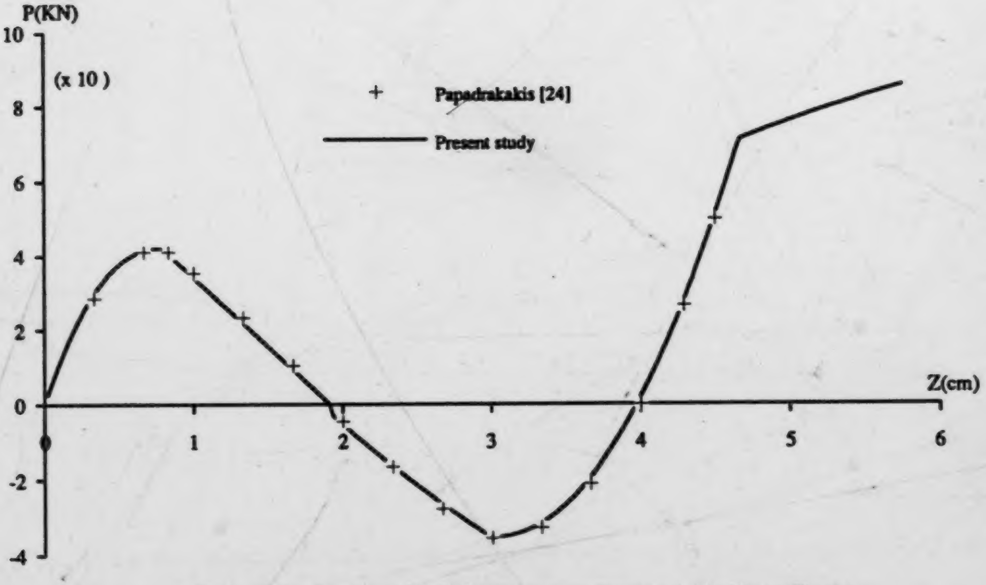


Figure (5.6m) Load-vertical displacement curve for node 1. (Nonlinear-Elastic)

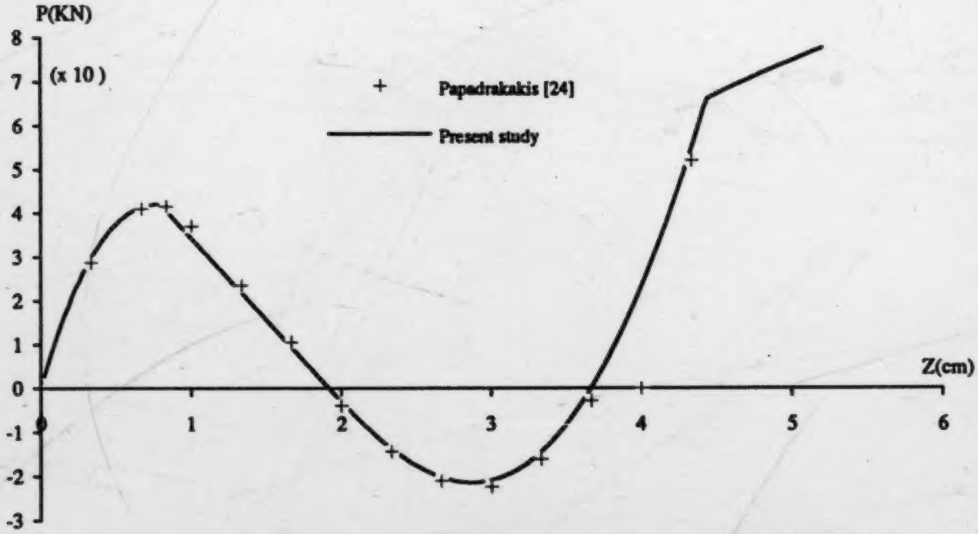


Figure (5.6n) Load-vertical displacement curve for node 1. (Elastic-perfectly plastic)

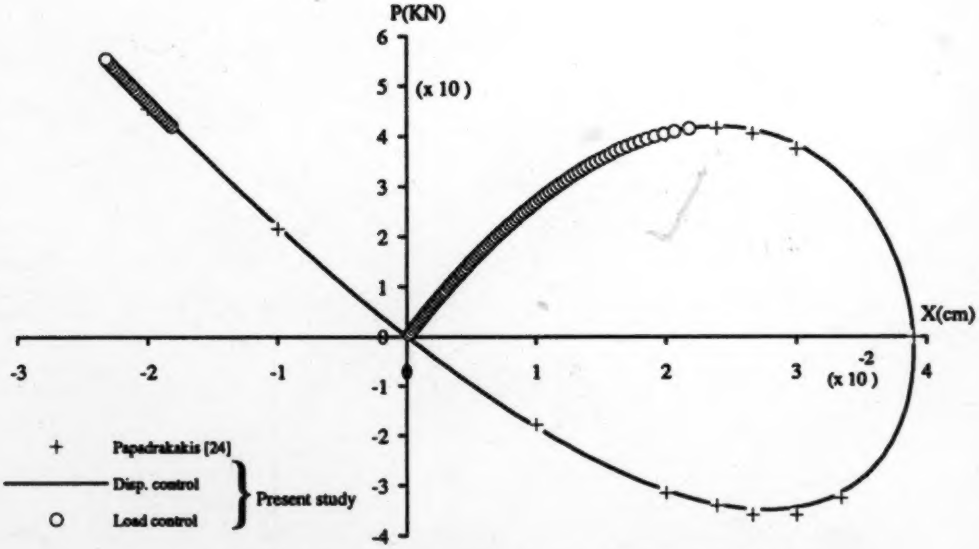


Figure (5.6 o) Load-horizontal displacement curve for node 2. (Linear-Elastic)

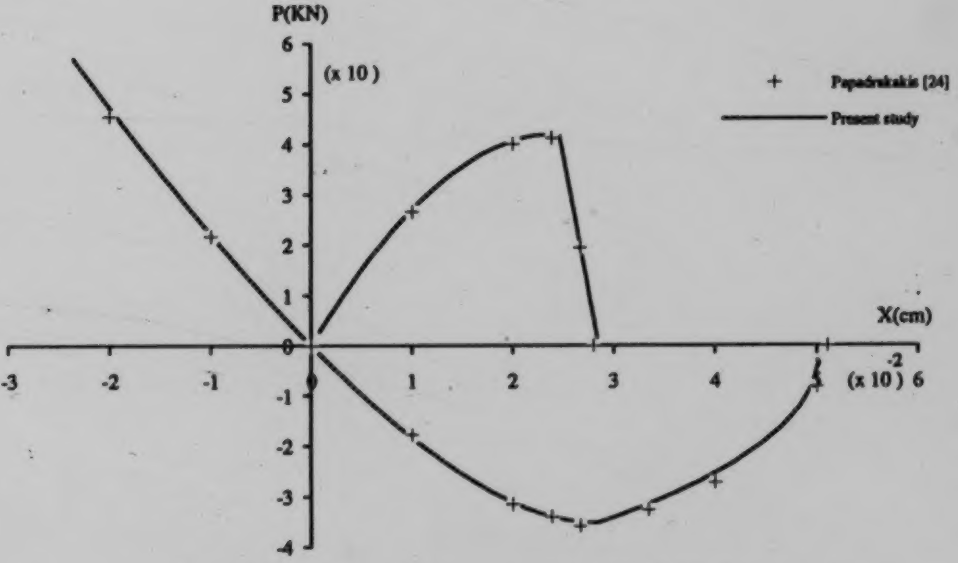


Figure (5.6p) Load-horizontal displacement curve for node 2. (Nonlinear-Elastic)

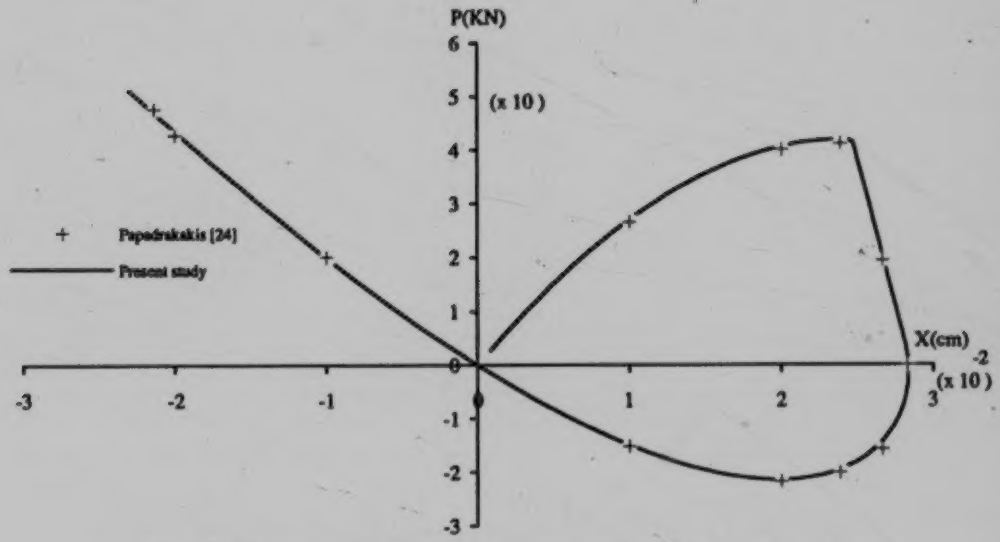


Figure (5.6q) Load-horizontal displacement curve for node 2. (Elastic-perfectly plastic)

Chapter 6

Concluding Remarks

6.1 Summary and Conclusions

In this work the numerical procedure Dynamic Relaxation (DR) has been adopted in an attempt to analyze problems involving large three-dimensional deformations of flexible rods. Since the deformations considered are large, these problems are governed by highly nonlinear equations which are difficult to solve analytically. This is further compounded by the wide variety of boundary condition which may be involved. As a result numerical procedure are usually required to obtain approximate solutions to such problems.

DR is an explicit iterative method developed for static analysis of structural mechanics problems. DR involves converting first the equilibrium equation into an equation of a damped motion, by artificially adding an acceleration term as well as a viscous damping term, and then integrating explicitly in time from the initial conditions until the transient dynamic response has damped out to the static solution, with equilibrium satisfied. The transient response obtained with this algorithm does not represent the real behavior of the structure, due to the fictitious mass and damping characteristics. However, since the force vector corresponds to the physical problem, the steady state part of the response, which satisfies the equilibrium equation, represents the static solution. Due to its explicit nature, the method is highly suitable for computations, since all quantities may be treated as vectors, eliminating the need for matrix manipulations of any kind. Therefore the method is easily programmable and has low storage requirements. Its explicit form makes it also ideal for case of large deformations with nonlinear geometric and material behavior.

To summarize the Comparison of NR with DR we could have the following.

1. Newton-Raphson
    - *Formulation.* More involved with the construction and inversion of the stiffness matrix.
    - *Memory requirement.* Needs large space to store stiffness matrix and its inversion.
    - *Computational time.* Takes less time.
    - *Computer code.* More difficult.
    - *Limit point tracing in post-buckling analysis.* Can not be traced accurately.
    - *Automation of the method.* Fully automated.
2. Dynamic Relaxation.
    - *Formulation.* Does not require the construction and inversion of the stiffness matrix, so DR is easier.
    - *Memory requirement.* Needs less space.
    - *Computational time.* Takes more time.
    - *Computer code.* Easier.
    - *Limit point tracing in post-buckling analysis.* Can be traced accurately.

- *Automation of the method.* Requires input of mass and damping values..

The automation of DR method may be difficult as it requires the input of certain parameters such as time step, mass and damping. The number of iterations required depends heavily on the mass and damping and time step values, with better values for them resulting in fewer iterations and consequently less computer time. Also, problems without twist involved show faster convergence behavior and require considerably shorter computer time. Even though the DR method is inferior compared to the NR method in the aspect of computational time, yet it offers an attractive alternative whenever both pre and post-buckling analyses need to be carried out for a given structure.

Since equilibrium solution obtained by DR would be regarded as asymptotically dynamically stable, the load control case DR does not pick up the unstable branch on which descending load accompanies increasing displacement. To get the whole load-displacement (stress-strain) curve as the stable solution we should use displacement control.

Apart from the *geometric nonlinearity* due to the inherently large deformation the structures can undergo, nonlinearity may be introduced into the problem through the material used. Most engineering structures are constructed of linearly elastic material, or more precisely materials used in their linearly elastic ranges. However increasing use is being made of materials which do not exhibit this type of linear elastic behavior. Such structures could also have a *material nonlinearity* which adds complexity to the problem. The numerical technique was also modified to include materials which exhibit inelastic behavior. The behavior of the structure were fully studied by means of the displacement analysis.

Inside each increment using DR the behavior of material considered on the specific path on the stress-strain curve and the irreversibility of inelastic strains has been taken into account only in passing from one increment of displacement stage to the subsequent one. The nonlinear governing equilibrium equations were solved using dynamic relaxation method, where the nonlinear behavior of the stress-strain relationships of members are followed in each iteration. For the inelastic material, since the behaviour of the structure is path-dependent,to get the right solution, the increment for applied load or applied displacement should be small, especially in the zones where the tangent modules has sudden and discontinuous changes. In chapter 5 the inclusion of both geometric and material nonlinearities in the analysis of elasto-plastic trusses was considered. In all cases for both linear and nonlinear materials where previous analytical, numerical or experimental results were available, the present method showed excellent agreement.

Constitutive relationships, which relate kinematics of the deformation to the forces and moments generated in the material, play an important role in applying the theory. Most commonly in the literature, the materials considered are assumed to be linearly elastic with the moments produced being related to the differences between the initial and final curvatures along the rod. For the material nonlinearity this work just considered truss structure in which axial force is the only load to be calculated from the constitutive law. Future work for material nonlinearity can be done in investigating the deformations with bending and twist involved. To generalize nonlinear deformations constitutive equations for moment versus curvature would be required.

Bibliography

[1] Antman S.S., *Nonlinear Problems of Elasticity.* Springer-Verlag, New York (1995).

[2] Argyris J.H., *Continua and discontinua* Proc. Conf. on Matrix Methods for strutral Mechanics, AFFDL-TR-66-80, Wrigth Patterson AFB, Ohio, 1965

[3] Barnes M.R., *Non-linear Numerical solution methods for Static and Dynamic Analysis of Tension Structures* Appl. Mech. Rev. Vol. 44, No. 7, July 1991, pp 38-56

[4] Bathe K. J., *Finite Element Procedures in Engineering Analysis.* 1982, pp. 311-313

[5] Batoz J.L. and Dhatt, *Incremental displacement algorithms for nonlinear problems* ,Int. J. numer. methods eng., **14** 1262-1266 (1979)

[6] Cannarozzi M., *Stationary and extremum Variational Formulations for the Elastostatics of cable networks.* Meccanica **20** (1985), 136-43.

[7] Coleman Bernard D.and Ellis H. Dill, *Flexure Waves in elastic Rods.* Acoust. Soc. Am., **91**(5), May (1993), 2663-2673.

[8] Crisfield M.A., *A fast incremental/iterative solution procedure that handles snap-through*, Comp. Struct., **13** 55-62 (1981)

[9] Dawe D. J., *Matrix and finite element displacement analysis of structures.* Oxford Engineering Science Series, 1984

[10] Ewing G.M., *Calculus of Variations with Applications.* W.W. Norton, New York (1969)

[11] Faulkner, M.G. and D.C. Stredulinsky *Nonlinear bending of inextensible thin rods under distributed and concentrated loads.* Transactions of the CSME 4(2), 77-82 (1976)

[12] Faulkner, M.G., A.W. Lipsett, K. El-Rayes, and D.L. Haberstock *On the use of vertical loops in reaction systems.* American Journal of Orthodontics and Dentofacial Orthopedics **99** (4), 328-336 (1991).

[13] Faulkner, M.G. and A.W. Lipsett, and V. Tam, *On the use of a segmental shooting technique for multiple solutions of planar elastica problems.* Computer methods in Applied Mechanics and Engineering 110, 221-236 (1993).

[14] Hangai Y.,and Kawamata, S. *Perturbation Method in the Analysis of Geometrically Nonlinear and Stability Problems* Advances in Computational Methods in Structural Mechanics and Design, UAH Press, Huntsville, Ala., 1972, pp. 473-489.

[15] Hangai Y.,and Kawamata, S. *Nonlinear Analysis of Space Frame and Span-through Buckling of reticulated Shell Structures.* Proceeding, 1971 IASS Pacific Symposium on Tension Structures and Space Frames, Architectural Institute of Japan, Tokyo , 1972, pp. 803-816.

[16] Landau L.D. and E.M. Lifshitz, *Theory of Elasticity.* Pergamon Press, 2^{th} edition, 1970

[17] Lipsett, A.W.,M.G. Faulkner, and K. El-Rayes *Large deformation analysis of orthodontic appliances.* Journal of Biomechanical Engineering **12**, 29-37 (1990).

[18] Lipsett, A.W., M.G.Faulkner, and V. Tam, *Multiple Solutions for Inextensible Arches.* Transactions of the CSME 17(1), 1-15 (1993).

[19] Love A.E.H., *A Treatise on the Mathematical Theory of Elasticity.* Dover Publication, 4^{th} edition, 1972

[20] Meek, J.L., and Tan, H.S., *Geometrically Nonlinear Analysis of Space Frames by an Incremental Iterative Technique* Computer Methods in Applied Mechanics and Engineering, Vol. 47, 1984, pp. 261-282.

[21] Meek, J.L., and Tan, H.S., *A Stiffness Matrix Extrapolation Strategy for Nonlinear Analysis* Computer Methods in Applied Mechanics and Engineering, Vol. 43, 1984, pp. 181-194.

[22] Papadrakakis M., *A Method for the Automatic Evaluation of the Dynamic Relaxation Parameters.* Computer Methods in Applied Mechanics and Engineering, **25**, (1981) pp. 35-48.

[23] Papadrakakis M., *Post-Buckling Analysis of Spatial Structures by Vector Iteration Methods.* Computers and structures, Vol 14, No. 5-6,1981, pp. 393-402.

[24] Papadrakakis M. , *Inelastic post-buckling analysis of trusses.* J. struct. div. ASCE. **109**, 2129-2147 (1982).

[25] Paradiso M., E. Reale and G. Tempesta, *Non Linear Post-buckling Analysis for Reticulated Dome Structure* Proc. IASS World Congress on Shell and Spatial Structures. Madrid, September (1979)

[26] Pian T.H.H. and P. Tong, *Variational formulation of finite-displacement analysis* in B.F. de Veubeke (ed.), High speed Computing of Elasic Structures, University of Liege, pp. 43-63, 1971

[27] Powell G. and J. Simons, *Improved iteration strategy for nonlinear structures* Internat. J. Numer. Methods. Engrg. 17 (1981) 1455-1467.

[28] Raboud, D., M.G. Faulkner, and A.W. Lipsett ***Multiple three-dimensional equilibrium solutions for cantiliever beams loaded by dead tip and uniform distributed loads.***International Journal of Non-Linear Mechanics **31**(3), 297-311 (1996a).

[29] Raboud, D., M.G. Faulkner, and A.W. Lipsett ***A segmental approach for large three-dimensional rod deformations.***International Journal of Solids and Structures **33** (8), 1137-1155 (1996b).

[30] Raboud, D., M.G. Faulkner, and A.W. Lipsett, and D.L. Haberstock, ***Three dimensional effects in reaction appliance design.*** American Journal of Orthodontics (1996).

[31] Ramm E., ***Strategies for Tracing the Nonlinear Response Near Limit Points.*** Nonlinear Finite Element Analysis in Structural Mechanics, W. Wunderlich, E.Stein and K. J. Bathe, eds., Springer-Verlag, 1981, pp. 63-89.

[32] Riks E., ***An incremental approach to the solution of snapping and buckling problems***, Int, J. Solids Struct, **15**, 529-551 (1979)

[33] Rothert H., Dickel T., and Renner D., ***Snap-Through Buckling of Reticulated Space Trusses*** Journal of the Structural Division, ASCE, Vol. 107, No. ST1, Proc. Paper 15973, Jan., 1981, pp. 129-143.

[34] Sack Ronald L., ***A Matrix structural analysis.*** PWS-KENT Publishing Company, 1989

[35] Sharifi P. and Popov E. P., ***Nonlinear buckling analysis of sandwich arches.*** J. Eng. Mech. Div. ASCE, **97** 1397-1412 (1971)

[36] Steigmann D.J. and M.G. Faulkner, ***Variational Theory for Spatial Rods.*** Journal of Elasticity, **33** (1993), 1-26.

[37] Steigmann D.J., *The Variational Structure of a Nonlinear Theory for Spatial Lattices*. Meccanica, **31** (1996), 441-455.

[38] Surana, K.S. and Sorem, R.M., *Geometrically Non-linear Formulation for Three Dimensional Curved Beam Elements with Large Rotation* International Journal for Numerical Methods in Engineering , Vol. 97, pp.637-644, 1971.

[39] Underwood P., *Dynamic Relaxation*. Computational methods for transient analysis, Edited by T. Bleystschko and T.J.R. Hughes, 1983, Elsevier science Publishers B.V., pp 245-265

[40] Wempner G.A., *Discrete approximations related to nonlinear theories of solids* ,Int, J. Solids Struct, **7** 1581-1599 (1971)

[41] Williiams F. W., *An Approach to the Nonlinear Behaviour of the Members of a Rigid Jointed Plane Framework with Finite Deflections*. Quart. J. Mech. Appl. Math 17, 451-469 (1979).

[42] Wood R.D. and O.C. Zienkiewicz, *Geometrically nonlinear finite element analysis of beams, frames, arches and axisymmetric shells*. Comput. and Structures 7 (1977) 725-735.

Appendix A

SOURCE CODE

This is the source code that is used for the numerical results.

```
C  AS DISPLACEMENT OR LOAD CONTROL FOR STRUCTURES WITH EITHER
C  RIGID OR HINGED UNCONSTRAINED WITH EITHER LINEAR ELASTIC OR
C  NONLINEAR ELASIT OR ELASTIC PERFECTLY PLASTIC OR ELASTIC
C  HARDENING PLASTIC.


C  TO GET THE RESULT FOR LINEAR ELASTIC ITYPEPROB=1
C  TO GET THE RESULT FOR NONLINEAR ELASTIC ITYPEPROB=2
C  TO GET THE RESULT FOR ELASTIC PERFECTLY PLASTIC ITYPEPROB=3
C  TO GET THE RESULT FOR ELASTIC HARDENING PLASTIC ITYPEPROB=4

C TO GET THE RIGHT VALUE FOR STRAIN IN THE ROD IN CASES 2,3 WE
C CALCULATE THE AVERAGE OF STRAIN IN THE ROD IN WHICH THE NO. OF
C NODE BETWEEN SHOULD BE ODD.


*     201 is the number of the nodes.
*     171 is the number of the rods.
*     99 is the maximum of the nnodbet+2
*     45 is the number of the constrained nodes
*     9 is the maximum number of the rods which come to the constrained node.
*     55 is the number of the unconstrained nodes
*     11 is the maximum number of the rods which come to the unconstrained
*     node.
```

```
  integer jstress(171),nrodstress
  Double precision r(171,99,3)
  Double precision lan(171,99)
  Double precision e2(171),e3(171)
  Double precision sigmay(171),x1,x2,length(171)

 double precision e(171),iz(171),area(171),gs(171),jx(171)
 double precision delta(171)
 double precision x(171,99,3)
  double precision fx(201),fy(201),fz(201),bool(171,3,3)
 double precision cu,mu
 double precision cv,mv,cw,mw
 double precision error,deltat
 double precision coefforce
 Double precision u0(55,3)
 integer icri(171),jcri(171),kcri(171),nf

  integer nnod,nrod,ncons,nuncons,conmat(171,2),ep(171)
 integer nnodbet(171)
 integer nrodsco(45),ntypeco(45),nnodcon(45)
 integer nrodcon(45,9)
 integer nrodsuc(55),ntypeun(55),nnodunc(55)
 integer nrodunc(55,11)
 integer nodload(55),nload
 integer nrodspre(55),ntypepre(55),nnodpre(55)
 integer nrodpre(55,11),npre

 character filein * 12, fileout * 12,fileout1 * 12

* asking the name of input file and output file

 print*, 'what is the name of intput file : '
 read*,filein
 print*, 'what is the name of output file : '
 read*,fileout
 print*, 'what is the name of output file for graphic : '
 read*,fileout1


*opening the input file and output file

 open (1,file=filein)
 open (2,file=fileout)
 open(3,file=fileout1)

* calling the needed subroutines
```

```
 call inputrod(icode,conmat,ncons,nuncons
 + ,e,iz,area,jx,gs,delta,nodload,fx,fy,fz
+ ,ntypeco,nrodcon,ntypeun,nrodunc,nnodcon
 + ,nnodunc,nrod,nnodbet,nrodsco,nrodsuc,nnod
 + ,cu,mu,cv,mv,cw,mw,error,deltat,x,bool,itype
+ ,icri,jcri,kcri,nf,coefforce,nload,nnodcri,nwgraph
+ ,nodefcri,ndir,nwconfig
 + ,nrodspre,ntypepre,nnodpre,nrodpre,npre,u0
 + ,sigmay,x1,x2,length,itypeprob,e2,e3
& ,jstress,nrodstress)

 Call DR(icode,lan,no,x,r,mu,mv,cw,mw
*            ,cu,cv,delta,e,area,jx,gs,iz,fx,fy,fz
*           ,nnod,error,deltat,bool,nrod,nnodbet,conmat
*           ,ncons,ntypeco,nrodsco,nrodcon,nnodcon
*           ,nuncons,ntypeun,nrodsuc,nrodunc,nnodunc
*           ,itype,icri,jcri,kcri,nf,coefforce,nodload,nload
*           ,nnodcri,nwgraph,nodefcri,ndir,nwconfig
 + ,nrodspre,ntypepre,nnodpre,nrodpre,npre,u0
 * ,sigmay,x1,x2,length,itypeprob,e2,e3
& ,jstress,nrodstress)

  call output(lan,no,r,nnod,nrod,nnodbet
 *   ,cu,mu,cv,mv,cw,mw,error,deltat,x,jf)


 stop
 end


 subroutine inputrod(icode,conmat,ncons,nuncons
 + ,e,iz,area,jx,gs,delta,nodload,fx,fy,fz
+ ,ntypeco,nrodcon,ntypeun,nrodunc,nnodcon
 + ,nnodunc,nrod,nnodbet,nrodsco,nrodsuc,nnod
 + ,cu,mu,cv,mv,cw,mw,error,deltat,x,bool,itype
+ ,icri,jcri,kcri,nf,coefforce,nload,nnodcri,nwgraph
+ ,nodefcri,ndir,nwconfig
 + ,nrodspre,ntypepre,nnodpre,nrodpre,npre,u0
 + ,sigmay,x1,x2,length,itypeprob,e2,e3
& ,jstress,nrodstress)

*    ======================================================

 character txt * 80

 integer jstress(171),nrodstress
```

```
      double precision coord(201,3),coefforce
      double precision etemp(171), inetemp(171), areatemp(171)
      double precision e(171),iz(171),area(171)
      double precision delta(171),length(171),sigmaytemp(171)
      double precision x(171,99,3),sigmay(171),x1,x2
      double precision fx(201),fy(201),fz(201)
      double precision cu,mu
      double precision cv,mv,cw,mw
      double precision error,deltat
      double precision bool(171,3,3),dx,dy,dz,lprime
      double precision gtemp(171),jtemp(171),gs(171),jx(171)
      Double precision u0(55,3),e2(171),e3(171),e2temp(171),e3temp(171)
      integer nnod,nrod,ncons,nuncons,conmat(171,2),ep(171)
      integer nnodbet(171)
      integer nrodsco(45),ntypeco(45),nnodcon(45)
      integer nrodcon(45,9)
      integer nrodsuc(55),ntypeun(55),nnodunc(55)
      integer nrodunc(55,11)
      integer nrodspre(55),ntypepre(55),nnodpre(55)
      integer nrodpre(55,11),npre
      integer nodload(55),icri(171),jcri(171),kcri(171),nf


      read(1,*) txt
      read(1,*) txt
      read(1,*) txt
      read(1,*) txt
      read(1,*) txt
      read(1,*) txt
      read(1,*) txt
      read(1,*) txt
      read(1,*) txt
      read(1,*) txt

      read(1,*) nnod,x1,x2,itypeprob

*     coord(i,1) is x of node i &  coord(i,2) is y of node i

      do 10 j = 1 , nnod
         read(1,*) i, (coord(j, k),k=1,3)
10    continue

* nproper is no. of properties in structure

      read(1,*) nproper

*reading material property
       read(1,*) txt
      do 20 j = 1 , nproper
          read(1,*) i, etemp(i), inetemp(i), areatemp(i)
     *          ,gtemp(i),jtemp(i),sigmaytemp(i)
     *          ,e2temp(i),e3temp(i)
20    continue

* nrod is no. of element in structure

      read(1,*) nrod

* reading material type ep(i),no. of first node and second node,

      do 30 j = 1 , nrod
          read(1,*) i, ep(i), conmat(i, 1), conmat(i, 2)
     +    ,nnodbet(i)
30    continue


      do 40 j = 1 , nrod
          e(j)=etemp(ep(j))
          area(j)=areatemp(ep(j))
          iz(j)=inetemp(ep(j))
          gs(j)=gtemp(ep(j))
          jx(j)=jtemp(ep(j))
        sigmay(j)=sigmaytemp(ep(j))
        e2(j)=e2temp(ep(j))
        e3(j)=e3temp(ep(j))
40    continue


      do 50 j = 1 , nrod
      length(j)=Dsqrt((coord(conmat(j,2),1)-coord(conmat(j,1),1))**2D0
     +          +(coord(conmat(j,2),2)-coord(conmat(j,1),2))**2D0
     +          +(coord(conmat(j,2),3)-coord(conmat(j,1),3))**2D0)
          delta(j)=length(j)/(nnodbet(j)+1D0)
          x(j,1,1)=coord(conmat(j,1),1)
          x(j,1,2)=coord(conmat(j,1),2)
          x(j,1,3)=coord(conmat(j,1),3)
          x(j,nnodbet(j)+2,1)=coord(conmat(j,2),1)
          x(j,nnodbet(j)+2,2)=coord(conmat(j,2),2)
          x(j,nnodbet(j)+2,3)=coord(conmat(j,2),3)
          do 60 i=2,nnodbet(j)+1
                x(j,i,1)=x(j,1,1)+(i-1D0)*(x(j,nnodbet(j)+2,1)
     +          -x(j,1,1))/(nnodbet(j)+1D0)
                x(j,i,2)=x(j,1,2)+(i-1D0)*(x(j,nnodbet(j)+2,2)
     +          -x(j,1,2))/(nnodbet(j)+1D0)
                x(j,i,3)=x(j,1,3)+(i-1D0)*(x(j,nnodbet(j)+2,3)
     +          -x(j,1,3))/(nnodbet(j)+1D0)
60        continue
50     continue

       do 31 j = 1,nrod

       dx=coord(conmat(j,2),1)-coord(conmat(j,1),1)
       dy=coord(conmat(j,2),2)-coord(conmat(j,1),2)
       dz=coord(conmat(j,2),3)-coord(conmat(j,1),3)
        if((dx.eq.0).and.(dz.eq.0)) then
           if(coord(conmat(j,2),2).gt.coord(conmat(j,1),2)) then
             bool(j,1,2)=1D0
             bool(j,2,1)=-1D0
             bool(j,3,3)=1D0
           else
             bool(j,1,2)=-1D0
             bool(j,2,1)=1D0
             bool(j,3,3)=1D0
           endif
       else
           bool(j,1,1)=dx/(delta(j)*(nnodbet(j)+1D0))
           bool(j,1,2)=dy/(delta(j)*(nnodbet(j)+1D0))
           bool(j,1,3)=dz/(delta(j)*(nnodbet(j)+1D0))
           lprime=sqrt(bool(j,1,1)**2D0+bool(j,1,3)**2D0)
           bool(j,2,1)=-(bool(j,1,1)*bool(j,1,2))/lprime
           bool(j,2,2)=lprime
           bool(j,2,3)=-bool(j,1,2)*bool(j,1,3)/lprime
           bool(j,3,1)=-bool(j,1,3)/lprime
           bool(j,3,3)=bool(j,1,1)/lprime
       endif
31     continue

       read(1,*) ncons,nuncons


       do 70 i = 1 , ncons

*  nrodsco(i) is the no. of rods which come to the same constraint
*  ntypeco(i) is the type of constraint

        read(1,*) nnodcon(i), ntypeco(i)

70     continue

       if (nuncons.ne.0) then
        do 90 i=1,nuncons

*   nrodsuc(i) is the no. of rods which comes to the same constraint
*  ntypeun(i) is the type of constraint

      read(1,*) nnodunc(i), ntypeun(i)

90    continue
     endif

     read(1,*) npre

     if (npre.ne.0) then
      do i=1,npre
       read(1,*) nnodpre(i), ntypepre(i),u0(i,1),u0(i,2),u0(i,3)
      enddo
     endif

     do i=1,npre
         ipre=0
          do j=1,nrod
           if (conmat(j,1).eq.nnodpre(i))then
            ipre=ipre+1
            nrodpre(i,ipre)=j
           endif
           if (conmat(j,2).eq.nnodpre(i))then
            ipre=ipre+1
            nrodpre(i,ipre)=j
           endif
        enddo
      nrodspre(i)=ipre
     enddo

     do 100 i=1,ncons
         icon=0
          do 110 j=1,nrod
           if (conmat(j,1).eq.nnodcon(i))then
            icon=icon+1
            nrodcon(i,icon)=j
           endif
           if (conmat(j,2).eq.nnodcon(i))then
            icon=icon+1
            nrodcon(i,icon)=j
           endif
110      continue
      nrodsco(i)=icon
100   continue


      do 130 i=1,nuncons
         iuncon=0
```

```
          do 120 j=1,nrod
           if (conmat(j,1).eq.nnodunc(i))then
            iuncon=iuncon+1
            nrodunc(i,iuncon)=j
           endif
           if (conmat(j,2).eq.nnodunc(i))then
            iuncon=iuncon+1
            nrodunc(i,iuncon)=j
           endif
120      continue
        nrodsuc(i)=iuncon
130    continue

      read(1,*) nload

* nodload(i) is the nod no. of the load

      do 140 j=1,nload
        read(1,*) nodload(j),fx(nodload(j)),fy(nodload(j))
     *          ,fz(nodload(j))
140   continue

      read(1,*) txt
      read(1,*) error,deltat
      read(1,*) txt
      read(1,*) nnodcri
      read(1,*) txt

      do j=1,nnodcri
        read(1,*) icri(j),jcri(j),kcri(j)
      enddo

      read(1,*) txt
       read(1,*)nf,coefforce,nwgraph,nwconfig
      read(1,*) txt
      read(1,*)nodefcri,ndir
      read(1,*) txt
      read(1,*) nrodstress
      read(1,*) txt

       do ks=1,nrodstress
        read(1,*) jstress(ks)
      enddo

      read(1,*) txt

      read(1,*) cu,mu
      read(1,*) cv,mv
read(1,*) cw,mw


return
end

  Subroutine DR(icode,lan,no,x,r,mu,mv,cw,mw
*              ,cu,cv,delta,e,area,jx,gs,iz,fx,fy,fz
*             ,nnod,error,deltat,bool,nrod,nnodbet,conmat
*             ,ncons,ntypeco,nrodsco,nrodcon,nnodcon
*             ,nuncons,ntypeun,nrodsuc,nrodunc,nnodunc
*             ,itype,icri,jcri,kcri,nf,coefforce,nodload,nload
*             ,nnodcri,nwgraph,nodefcri,ndir,nwconfig
+ ,nrodspre,ntypepre,nnodpre,nrodpre,npre,u0
* ,sigmay,x1,x2,length,itypeprob,e2,e3
& ,jstress,nrodstress)

integer jstress(171),nrodstress
 Double precision e2(171),e3(171)
 Double precision epsilonold(171,99),sigmaold(171,99)
 Double precision epsilonnew(171,99),sigmanew(171,99)
 Double precision tempaveeps(1000,171),tempavsigma(1000,171)
 Double precision sigmay(171),x1,x2,length(171),pi
 Double precision tempcri(10000,171),fcri(10000)
     Double precision x(171,99,3),r(171,99,3),rr(171,99,3)
     Double precision mu,mv,cu,cv,delta(171),e(171),rmaxu,rmaxv
     Double precision area(171),iz(171),fx(201),gs(171),jx(171)
     Double precision fy(201),fz(201),udotp(171,99,3),bool(171,3,3)
     Double precision udotc(171,99,3),deltat,u(171,99,3)
     Double precision error,vu,vv,lan(171,99)
     Double precision t(171,99,3,3),cw,mw,teta(171,99,3)
     Double precision rmax1,rmax2,teta0,dteta
    double precision coefforce,fx0(201),fy0(201),fz0(201)
     Double precision r0(171,99,3),u0(55,3),f(171,99,3)
  Double precision tempepsilon(10000,171),tempsigma(10000,171)
  integer icri(171),jcri(171),kcri(171),nf,nload
 integer icodeold(171,99)
 integer icodenew(171,99)
integer ncons,nuncons,conmat(171,2)
integer nrodsco(45),ntypeco(45),nnodcon(45)
integer nrodcon(45,9)
integer nrodsuc(55),ntypeun(55),nnodunc(55)
integer nrodunc(55,11)
integer nodload(55)
integer nnodbet(171),nrod
integer nrodspre(55),ntypepre(55),nnodpre(55)
integer nrodpre(55,11),npre
```

```
      print*,'what is the no. of iterations for error to be writen'
      read*,iwrite

     pi=4D0*Atan(1D0)


          itr=0
       kwrite=0
          no=0
       nwrite=1
          vu=deltat*cu
         vv=deltat*cv
         vw=deltat*cw
      do 1000 j = 1 , nrod
          do 1005 i=1,nnodbet(j)+2
                   u(j,i,1)=0D0
                   u(j,i,2)=0D0
                   u(j,i,3)=0D0
1005      continue
1000    continue


     do jl=1,nload
       fx0(nodload(jl))=fx(nodload(jl))
       fy0(nodload(jl))=fy(nodload(jl))
       fz0(nodload(jl))=fz(nodload(jl))
     enddo

6000       do 9000 jf=1,nf

     do jl=1,nload
       fx(nodload(jl))=fx0(nodload(jl))*(1D0+(jf-1d0)*coefforce)
       fy(nodload(jl))=fy0(nodload(jl))*(1D0+(jf-1d0)*coefforce)
       fz(nodload(jl))=fz0(nodload(jl))*(1D0+(jf-1d0)*coefforce)
c      fx(nodload(jl))=fx0(nodload(jl))*Sin(2D0*pi*jf/nf)
c      fy(nodload(jl))=fy0(nodload(jl))*Sin(2D0*pi*jf/nf)
c      fz(nodload(jl))=fz0(nodload(jl))*Sin(2D0*pi*jf/nf)
     enddo

      do j=1,npre
        do kd=1,3
             do k=1,nrodspre(j)
                nr=nrodpre(j,k)
                nj=nnodbet(nrodpre(j,k))+2
              if(conmat(nrodpre(j,k),1).eq.nnodpre(j)) then
             r0(nr,1,kd)=x(nr,1,kd)
     &       +u0(j,kd)*(1D0+(jf-1d0)*coefforce)
```

```
c             r0(nr,1,kd)=x(nr,1,kd)
c    &       +u0(j,kd)**Sin(2D0*pi*jf/nf)
            else
              r0(nr,nj,kd)=x(nr,nj,kd)
   &       +u0(j,kd)*(1D0+(jf-1d0)*coefforce)

c           r0(nr,nj,kd)=x(nr,nj,kd)
c    &      +u0(j,kd)**Sin(2D0*pi*jf/nf)

            endif
        enddo
      enddo
     enddo

1900     Do 1010 j=1 , nrod
          do 1015 i=1,nnodbet(j)+2
                r(j,i,1)=x(j,i,1)+u(j,i,1)
                r(j,i,2)=x(j,i,2)+u(j,i,2)
                r(j,i,3)=x(j,i,3)+u(j,i,3)
1015       continue
1010     Continue

     do jp=1,npre
       do kd=1,3
           do k=1,nrodspre(jp)
              nr=nrodpre(jp,k)
              nj=nnodbet(nrodpre(jp,k))+2
            if(conmat(nrodpre(jp,k),1).eq.nnodpre(jp)) then
              r(nr,1,kd)=r0(nr,1,kd)
            else
              r(nr,nj,kd)=r0(nr,nj,kd)
            endif
        enddo
       enddo
     enddo

        Call calcres(x,lan,r,delta,e,area,jx,gs,iz,fx,fy,fz
    *          ,nnod,rr,bool,nrod,nnodbet,conmat
    *          ,ncons,ntypeco,nrodsco,nrodcon,nnodcon
    *          ,nuncons,ntypeun,nrodsuc,nrodunc,nnodunc
    *          ,t,teta0,teta,itype
    * ,nrodspre,ntypepre,nnodpre,nrodpre,npre
    * ,sigmay,x1,x2,length,itypeprob
    *          ,epsilonold,sigmaold,f,icodeold
    *          ,epsilonnew,sigmanew,icodenew,iconverged,e2,e3)

      rmaxu=0D0
        rmaxv=0D0
```

```
          rmaxw=0D0

        Do 1020 j=1 , nrod
            do 1025 i=1,nnodbet(j)+2
              If (Abs(rr(j,i,1)).gt.rmaxu) Then
                rmaxu=Abs(rr(j,i,1))
                jju=j
               iiu=i
              End If
              If (Abs(rr(j,i,2)).gt.rmaxv) Then
                rmaxv=Abs(rr(j,i,2))
                jjv=j
               iiv=i
              End If
              If (Abs(rr(j,i,3)).gt.rmaxw) Then
                rmaxw=Abs(rr(j,i,3))
                jjw=j
               iiw=i
              End If
1025          continue
1020      Continue

c         write(*,*)'rmax u,v,w=',rmaxu,rmaxv,rmaxw


        if(nwrite.eq.iwrite) then
          write(*,*)'rmax u,v=',rmaxu,rmaxv
          write(*,*)'rmax w=',rmaxw
         nwrite=0
        endif

            If (rmaxv.gt.rmaxu) Then
                rmax1=rmaxv
             else
                rmax1=rmaxu
             End If

            If (rmaxw.gt.rmax1) Then
                rmax2=rmaxw
             else
                rmax2=rmax1
             End If

           If (rmax2.lt.error) Go To 1950
            If (itr.eq.0) Then

            Do 1030 j=1,nrod
              do 1035 i=1, nnodbet(j)+2
          udotc(j,i,1)=(deltat/(2D0*mu))*rr(j,i,1)
          udotc(j,i,2)=(deltat/(2D0*mv))*rr(j,i,2)
          udotc(j,i,3)=(deltat/(2D0*mw))*rr(j,i,3)
1035     continue
1030    Continue
       Else

       do 1045 j=1,nrod
         Do 1040 i=1 , nnodbet(j)+2
           udotc(j,i,1)=((2d0-vu)/(2d0+vu))*udotp(j,i,1)
   *                +(2d0*deltat/(2d0+vu))*rr(j,i,1)/mu
          udotc(j,i,2)=((2d0-vv)/(2d0+vv))*udotp(j,i,2)
   *                +(2d0*deltat/(2d0+vv))*rr(j,i,2)/mv
          udotc(j,i,3)=((2d0-vw)/(2d0+vw))*udotp(j,i,3)
   *                +(2d0*deltat/(2d0+vw))*rr(j,i,3)/mw
1040     Continue
1045    continue
       End If
       do 1055 j=1,nrod
       Do 1050 i=1 ,nnodbet(j)+2
         u(j,i,1)=u(j,i,1)+udotc(j,i,1)*deltat
         u(j,i,2)=u(j,i,2)+udotc(j,i,2)*deltat
         u(j,i,3)=u(j,i,3)+udotc(j,i,3)*deltat
        udotp(j,i,1)=udotc(j,i,1)
        udotp(j,i,2)=udotc(j,i,2)
        udotp(j,i,3)=udotc(j,i,3)
1050    Continue
1055   continue

       itr=itr+1
       no=no+1
     nwrite=nwrite+1
       Go To 1900

1950    write(*,*)'converged',jf
     iconverged=iconverged+1

    if(MOD(jf,nwgraph).eq.0) then
     kwrite=kwrite+1

     do ks=1,nrodstress
     tempaveeps(kwrite,jstress(ks))=0d0
     tempavsigma(kwrite,jstress(ks))=0d0
     nj=nnodbet(jstress(ks))+2
     Do i=2 , nj-1,2
      tempaveeps(kwrite,jstress(ks))=tempaveeps(kwrite,jstress(ks))
   &                       +epsilonnew(jstress(ks),i)
        tempavsigma(kwrite,jstress(ks))=tempavsigma
     &                (kwrite,jstress(ks))+sigmanew(jstress(ks),i)
        enddo
        tempaveeps(kwrite,jstress(ks))=
     &             (tempaveeps(kwrite,jstress(ks))*2d0)/(nj*1d0-1d0)
        tempavsigma(kwrite,jstress(ks))=
     &             (tempavsigma(kwrite,jstress(ks))*2d0)/(nj*1d0-1d0)

       enddo


          do kw=1,nnodcri
           tempcri(kwrite,kw)=x(icri(kw),jcri(kw),kcri(kw))-
     &                        r(icri(kw),jcri(kw),kcri(kw))

          enddo

          if(ndir.eq.1)then
           fcri(kwrite)=-fx(nodefcri)
          elseif(ndir.eq.2)then
           fcri(kwrite)=-fy(nodefcri)
          elseif(ndir.eq.3)then
           fcri(kwrite)=-fz(nodefcri)
          endif

c         write(2,*) 'kwrite',kwrite

       do jwr=1,nnodcri
cc          write(2,*) jwr,',',fcri(kwrite)
cc     &              ,',',tempcri(kwrite,jwr)

c          write(*,*) jwr,',',fcri(kwrite)
c      &,',',tempcri(kwrite,jwr)
       enddo



       do jwr=1,nnodcri
c          write(*,*) jwr,',',tempepsilon(kwrite,jwr)
c      &,',',tempsigma(kwrite,jwr)
       enddo


       endif

       if((MOD(jf,nwconfig).eq.0).or.(jf.eq.1)) then

c          if((jf.eq.3).or.(jf.eq.4).or.(jf.eq.5).or.(jf.eq.25)
c      &         .or.(jf.eq.26).or.(jf.eq.27)) then
```

```
ccc          call output(lan,no,r,nnod,nrod,nnodbet
ccc    *             ,cu,mu,cv,mv,cw,mw,error,deltat,x,jf)
c            write(2,*)'No. of force',jf
c         endif
       endif

       do j=1,nrod
        nj=nnodbet(j)+2
        Do i=1 , nj
ccc            write(2,*) 'lan,f1(',j,i,')=',lan(j,i),f(j,i,1)
        enddo
       enddo

       do j=1,nrod
        nj=nnodbet(j)+2
        Do i=1 , nj
ccc            write(2,*) 'epsilon,sigma(',j,i,')='
ccc    &              ,epsilonnew(j,i),sigmanew(j,i)
        enddo
       enddo



       do j=1,nrod
        nj=nnodbet(j)+2
        Do i=1 , nj
         epsilonold(j,i)=epsilonnew(j,i)
         sigmaold(j,i)=sigmanew(j,i)
         icodeold(j,i)=icodenew(j,i)
        Enddo
       enddo

9000  continue

       do j=1,nnodcri
         write(3,*)'j',j
        do i=1,kwrite
         write(3,*) tempcri(i,j),',', fcri(i)
        enddo
       enddo


c       if((itypeprob.eq.2) .or.(itypeprob.eq.3)) then
       do ks=1,nrodstress
        write(3,*)'average strain rod',jstress(ks)
        do i=1,kwrite
          write(3,*) tempaveeps(i,jstress(ks))/1d3,',',
     &              tempavsigma(i,jstress(ks))/1d1
```

```
        enddo
       enddo
c       elseif

c       do j=1,nnodcri
c         write(3,*)'j stress',j
c        do i=1,kwrite
c         write(3,*) tempepsilon(i,j),',', tempsigma(i,j)
c        enddo
c       enddo

c      endif

       Return
       End

        Subroutine constitutive2(lan,e,area,length
     *            ,iz,nrod,nnodbet,f,x1,x2,sigmay,iprint
     *           ,epsilonold,sigmaold,icodeold
     *           ,epsilonnew,sigmanew,icodenew)

       Double precision epsilonold(171,99),sigmaold(171,99)
       Double precision f(171,99,3),iz(171),e(171),area(171)
       Double precision epsilonnew(171,99),sigmanew(171,99)
       Double precision lan(171,99),pi
       Double precision sigmay(171),length(171)
       Double precision epsilonyp(171),epsilonyn(171)
       Double precision epsilonup,epsilondown
       integer nnodbet(171),nrod,icodeold(171,99),icodenew(171,99)

       pi=4D0*Atan(1D0)

       do j=1,nrod
       epsilonyp(j)=sigmay(j)/e(j)
       epsilonyn(j)=-sigmay(j)/e(j)
       nj=nnodbet(j)+2
       Do i=1 , nj
        epsilonnew(j,i)=lan(j,i)-1D0

         if(epsilonnew(j,i).eq.epsilonold(j,i)) then
C  The stress-strain state is the same as before.

         sigmanew(j,i)=sigmaold(j,i)

         elseif(epsilonnew(j,i).ge.epsilonyp(j)) then
C  The stress-strain state is on the line sigmay

          sigmanew(j,i)=sigmay(j)
          elseif(epsilonnew(j,i).lt.epsilonyn(j)) then
C The stress-strain state is on the line -sigmay
           sigmanew(j,i)=-sigmay(j)

          else
C The stress-strain state is on the elastic part.

           sigmanew(j,i)=e(j)*epsilonnew(j,i)

          endif

       f(j,i,1)=sigmanew(j,i)*area(j)
        Enddo
       enddo


       Return
       End


        Subroutine constitutive3(lan,e,area,length
     *            ,iz,nrod,nnodbet,f,x1,x2,sigmay,iprint
     *           ,epsilonold,sigmaold,icodeold
     *           ,epsilonnew,sigmanew,icodenew)


        Double precision epsilonold(171,99),sigmaold(171,99)
        Double precision f(171,99,3),iz(171),e(171),area(171)
        Double precision epsilonnew(171,99),sigmanew(171,99)
        Double precision lan(171,99),pi
        Double precision sigmay(171),length(171)
        Double precision epsilonyp(171),epsilonyn(171)
        Double precision epsilonup,epsilondown
        integer nnodbet(171),nrod,icodeold(171,99),icodenew(171,99)


        pi=4D0*Atan(1D0)

        do j=1,nrod
        epsilonyp(j)=sigmay(j)/e(j)
        epsilonyn(j)=-sigmay(j)/e(j)
        nj=nnodbet(j)+2
        Do i=1 , nj
         epsilonnew(j,i)=lan(j,i)-1D0

         if(epsilonnew(j,i).eq.epsilonold(j,i)) then
```

```
C The stress-strain state is the same as before.

        sigmanew(j,i)=sigmaold(j,i)
        icodenew(j,i)=icodeold(j,i)
       elseif(epsilonnew(j,i).gt.epsilonold(j,i)) then
C Strain is increasing

        if(icodeold(j,i).eq.1) then
C The stress-strain state is not on the normal curve.

         epsilonup=(sigmay(j)-sigmaold(j,i))/e(j)
     &              +epsilonold(j,i)
         if(epsilonnew(j,i).lt.epsilonup) then
C The stress-strain state is under the line  sigmay

          sigmanew(j,i)=e(j)*(epsilonnew(j,i)-epsilonold(j,i))
     &              +sigmaold(j,i)
          icodenew(j,i)=1
         else


C The stress-strain state is on the line sigmay

          sigmanew(j,i)=sigmay(j)
          icodenew(j,i)=0
         endif
        else
         if(sigmaold(j,i).eq.sigmay(j)) then
          sigmanew(j,i)=sigmay(j)
          icodenew(j,i)=0
C The stress-strain state is on the normal curve.

         elseif(epsilonnew(j,i).ge.epsilonyp(j)) then
C The stress-strain state is on the line sigmay

          sigmanew(j,i)=sigmay(j)
          icodenew(j,i)=0
         elseif(epsilonnew(j,i).lt.epsilonyn(j)) then
C The stress-strain state is on the line -sigmay

          sigmanew(j,i)=e(j)*(epsilonnew(j,i)-epsilonold(j,i))
     &               +sigmaold(j,i)
          icodenew(j,i)=1
         else
C The stress-strain state is on the elastic part.

          sigmanew(j,i)=e(j)*epsilonnew(j,i)
          icodenew(j,i)=0
```

```
           endif
          endif

        else
C Strain is decreasing

          if(icodeold(j,i).eq.1) then
C The stress-strain state is not on the normal curve.

          epsilondown=(-sigmay(j)-sigmaold(j,i))/e(j)
     &               +epsilonold(j,i)
          if(epsilonnew(j,i).ge.epsilondown) then
C The stress-strain state is upper the line -sigmay

           sigmanew(j,i)=e(j)*(epsilonnew(j,i)-epsilonold(j,i))
     &               +sigmaold(j,i)
           icodenew(j,i)=1
          else
C The stress-strain state is on the line -sigmay

            sigmanew(j,i)=-sigmay(j)
            icodenew(j,i)=0
           endif
          else

           if(sigmaold(j,i).eq.-sigmay(j)) then
            sigmanew(j,i)=-sigmay(j)
            icodenew(j,i)=0

C The stress-strain state is on the normal curve.

          elseif(epsilonnew(j,i).lt.epsilonyn(j)) then
C The stress-strain state is on the line -sigmay

           sigmanew(j,i)=-sigmay(j)
           icodenew(j,i)=0
          elseif(epsilonnew(j,i).lt.epsilonyp(j)) then
C The stress-strain state is on the elastic part.

            sigmanew(j,i)=e(j)*epsilonnew(j,i)
            icodenew(j,i)=0
           else
C The stress-strain state is on the sigmay

            sigmanew(j,i)=e(j)*(epsilonnew(j,i)-epsilonold(j,i))
     &               +sigmaold(j,i)
            icodenew(j,i)=1
           endif
          endif
         endif

        f(j,i,1)=sigmanew(j,i)*area(j)
         Enddo
        enddo


        Return
        End



         Subroutine constitutive4(lan,e,area,length
     *            ,iz,nrod,nnodbet,f,x1,x2,sigmay,iprint
     *            ,epsilonold,sigmaold,icodeold
     *            ,epsilonnew,sigmanew,icodenew,iconverged,e2,e3)


        Double precision epsilonold(171,99),sigmaold(171,99)
        Double precision f(171,99,3),iz(171),e(171),area(171)
        Double precision epsilonnew(171,99),sigmanew(171,99)
        Double precision lan(171,99),pi,e2(171),e3(171)
        Double precision sigmay(171),length(171)
        Double precision epsilonyp(171),epsilonyn(171)
        Double precision epsilonup,epsilondown

        integer nnodbet(171),nrod,icodeold(171,99),icodenew(171,99)

        pi=4D0*Atan(1D0)

        do j=1,nrod
        epsilonyp(j)=sigmay(j)/e(j)
        epsilonyn(j)=-sigmay(j)/e(j)
        nj=nnodbet(j)+2
        Do i=1 , nj
         epsilonnew(j,i)=lan(j,i)-1D0

c         if(iconverged.gt.340) then
c            print*,'icodenew,icodeold',
c     &              icodenew(j,i),icodeold(j,i)
c            print*,'sigmanew,sigmaold',
c     &              sigmanew(j,i),sigmaold(j,i)
c            print*,'epsilonnew,epsilonold',
c     &              epsilonnew(j,i),epsilonold(j,i)
c             print*,'j,i',j,i
c         endif
```

```
      if(epsilonnew(j,i).eq.epsilonold(j,i)) then
C The stress-strain state is the same as before.

         sigmanew(j,i)=sigmaold(j,i)
       icodenew(j,i)=icodeold(j,i)
      elseif(epsilonnew(j,i).gt.epsilonold(j,i)) then
C Strain is increasing

        if(icodeold(j,i).eq.1) then
C The stress-strain state is not on the normal curve.

          epsilonup=(sigmay(j)-sigmaold(j,i)+e(j)*epsilonold(j,i)
     &           -e2(j)*epsilonyp(j))/(e(j)-e2(j))

          if(epsilonnew(j,i).lt.epsilonup) then
C The stress-strain state is under the line  sigmay

          sigmanew(j,i)=e(j)*(epsilonnew(j,i)-epsilonold(j,i))
     &              +sigmaold(j,i)
          icodenew(j,i)=1
         else


C The stress-strain state is on the line sigmay
           sigmanew(j,i)=e2(j)*(epsilonnew(j,i)-epsilonyp(j))
     &               +sigmay(j)

           icodenew(j,i)=4
         endif
        else

        if(icodeold(j,i).eq.4) then

           sigmanew(j,i)=e2(j)*(epsilonnew(j,i)-epsilonyp(j))
     &               +sigmay(j)

           icodenew(j,i)=4

C The stress-strain state is on the normal curve.

        elseif(epsilonnew(j,i).ge.epsilonyp(j)) then
C The stress-strain state is on the line sigmay
           sigmanew(j,i)=e2(j)*(epsilonnew(j,i)-epsilonyp(j))
     &               +sigmay(j)

           icodenew(j,i)=4
         elseif(epsilonnew(j,i).lt.epsilonyn(j)) then
```

```
C The stress-strain state is on the line -sigmay

        sigmanew(j,i)=e(j)*(epsilonnew(j,i)-epsilonold(j,i))
     &              +sigmaold(j,i)
        icodenew(j,i)=1
       else
C The stress-strain state is on the elastic part.

        sigmanew(j,i)=e(j)*epsilonnew(j,i)
        icodenew(j,i)=0
       endif
      endif

     else
C Strain is decreasing

      if(icodeold(j,i).eq.1) then

c     if(iconverged.gt.340) then
c       print*,'step 1'
c     endif

C The stress-strain state is not on the normal curve.

       epsilondown=(-sigmay(j)-sigmaold(j,i)+e(j)*epsilonold(j,i)
     &          -e3(j)*epsilonyn(j))/(e(j)-e3(j))

       if(epsilonnew(j,i).gt.epsilondown) then
C The stress-strain state is upper the line -sigmay

c     if(iconverged.gt.340) then
c       print*,'step 2'
c     endif


        sigmanew(j,i)=e(j)*(epsilonnew(j,i)-epsilonold(j,i))
     &              +sigmaold(j,i)
        icodenew(j,i)=1
       else
C The stress-strain state is on the line -sigmay

        sigmanew(j,i)=e3(j)*(epsilonnew(j,i)-epsilonyn(j))
     &                -sigmay(j)
        icodenew(j,i)=2

c     if(iconverged.gt.340) then
c       print*,'step 3'
c     endif
```

```
         endif
c       if(iconverged.gt.340) then
c         print*,'step 4'
c       endif

        else
         if(icodeold(j,i).eq.2) then

c       if(iconverged.gt.340) then
c         print*,'step 5'
c       endif

          sigmanew(j,i)=e3(j)*(epsilonnew(j,i)-epsilonyn(j))
     &               -sigmay(j)
          icodenew(j,i)=2


C The stress-strain state is on the normal curve.

         elseif(epsilonnew(j,i).lt.epsilonyn(j)) then
C The stress-strain state is on the line -sigmay
          sigmanew(j,i)=e3(j)*(epsilonnew(j,i)-epsilonyn(j))
     &               -sigmay(j)
          icodenew(j,i)=2

c       if(iconverged.gt.340) then
c         print*,'step 6'
c       endif

         elseif(epsilonnew(j,i).lt.epsilonyp(j)) then
C The stress-strain state is on the elastic part.

          sigmanew(j,i)=e(j)*epsilonnew(j,i)
          icodenew(j,i)=0

c       if(iconverged.gt.340) then
c         print*,'step 7'
c       endif

         else
C The stress-strain state is on the sigmay

          sigmanew(j,i)=e(j)*(epsilonnew(j,i)-epsilonold(j,i))
     &               +sigmaold(j,i)
          icodenew(j,i)=1

c       if(iconverged.gt.340) then
```

```
c         print*,'step 8'
c       endif

          endif
         endif
        endif

       f(j,i,1)=sigmanew(j,i)*area(j)
        Enddo
       enddo


       Return
       End


         Subroutine calcres(x,lan,rg,delta,e,area,jx,gs,iz,fx,fy,fz
     *          ,nnod,rr,bool,nrod,nnodbet,conmat
     *          ,ncons,ntypeco,nrodsco,nrodcon,nnodcon
     *          ,nuncons,ntypeun,nrodsuc,nrodunc,nnodunc
     *          ,t,teta0,teta,itype
     * ,nrodspre,ntypepre,nnodpre,nrodpre,npre
     * ,sigmay,x1,x2,length,itypeprob
     *          ,epsilonold,sigmaold,f,icodeold
     *          ,epsilonnew,sigmanew,icodenew,iconverged,e2,e3)


        Double precision epsilonnew(171,99),sigmanew(171,99)
        integer icodenew(171,99)
       Double precision e2(171),e3(171)
        Double precision epsilonold(171,99),sigmaold(171,99)
        Double precision sigmay(171),x1,x2,length(171)
           Double precision rg(171,99,3),delta(171),e(171),area(171)
           Double precision gs(171),jx(171)
           Double precision iz(171),rrp(171,99,3)
           Double precision fx(201),fy(201),fz(201),rr(171,99,3)
           Double precision bool(171,3,3),r(171,99,3),x(171,99,3)
           Double precision rp(171,99,3),tetap(171,99,3)
           Double precision tetapp(171,99,3)
           Double precision t(171,99,3,3),lan(171,99)
           Double precision teta(171,99,3),sum(3)
        Double precision suml(3),teta0
           Double precision tp(171,99,3,3),f(171,99,3),fp(171,99,3)
        Double precision kapa(171,99,3),kapap(171,99,3)
        integer icodeold(171,99)
         integer ncons,nuncons,conmat(171,2)
         integer nrodsco(45),ntypeco(45),nnodcon(45)
        integer nrodcon(45,9)
        integer nrodsuc(55),ntypeun(55),nnodunc(55)
        integer nrodunc(55,11)
        integer nodload(55)
        integer nnodbet(171),nrod
      integer nrodspre(55),ntypepre(55),nnodpre(55)
      integer nrodpre(55,11),npre

             Call rprime(x,bool,rg,r,nnod,delta,rp,nrod,nnodbet)

             Call lanteta(rp,lan,nnod
     *               ,teta,nrod,nnodbet,conmat
     *               ,ncons,ntypeco,nrodsco,nrodcon,nnodcon
     *               ,nuncons,ntypeun,nrodsuc,nrodunc,nnodunc
     *               ,delta,teta0,itype
     + ,nrodspre,ntypepre,nnodpre,nrodpre,npre)

             Call trans(rp,lan,nrod,nnodbet,teta,t)

              Call tetaprime(lan,r,nnod,delta,tetap,tetapp,teta
     *                ,nrod,nnodbet)

             Call kapaprime(kapa,kapap,nrod,nnodbet
     *               ,teta,tetap,tetapp)

             Call calcf(lan,nnod,e,kapa,kapap
     *               ,jx,gs,area,nrod,nnodbet
     *               ,iz,delta,f,sigmay,x1,x2,length
     *               ,itypeprob
     *          ,epsilonold,sigmaold,icodeold
     *          ,epsilonnew,sigmanew,icodenew,iconverged,e2,e3)

             Call fprime(f,fp,delta,nrod,nnodbet)

             Call transprime(nrod,nnodbet,teta,tetap,tp)

        do 2005 j=1,nrod
             nj=nnodbet(j)+2
              Do 2000 i=2 , nj-1
              do 100 k=1,3
                 rrp(j,i,k)=fp(j,i,1)*t(j,i,1,k)+fp(j,i,2)*t(j,i,2,k)+
     *                 fp(j,i,3)*t(j,i,3,k)+f(j,i,1)*tp(j,i,1,k)+
     *                 f(j,i,2)*tp(j,i,2,k)+f(j,i,3)*tp(j,i,3,k)
100           continue
              do 120 k=1,3
                 rr(j,i,k)=rrp(j,i,1)*bool(j,1,k)+rrp(j,i,2)*bool(j,2,k)+
     *                rrp(j,i,3)*bool(j,3,k)
120           continue
```

```
2000      Continue
2005   continue

       do 4050 j=1,nuncons
          do 130 l=1,3
           sum(l)=0D0
130       continue
            do 4060 k=1,nrodsuc(j)
             nr=nrodunc(j,k)
             nj=nnodbet(nrodunc(j,k))+2
             if(conmat(nrodunc(j,k),1).eq.nnodunc(j)) then
             do 140 jj=1,3
               suml(jj)=f(nr,1,1)*t(nr,1,1,jj)+f(nr,1,2)*t(nr,1,2,jj)+
     *           f(nr,1,3)*t(nr,1,3,jj)
140          continue
             do 150 jj=1,3
              sum(jj)=sum(jj)+(suml(1)*bool(nr,1,jj)+
     *            suml(2)*bool(nr,2,jj)+suml(3)*bool(nr,3,jj))
150          continue
             else
             do 160 jj=1,3
              suml(jj)=f(nr,nj,1)*t(nr,nj,1,jj)
     *          +f(nr,nj,2)*t(nr,nj,2,jj)+f(nr,nj,3)*t(nr,nj,3,jj)
160          continue
             do 170 jj=1,3
              sum(jj)=sum(jj)-(suml(1)*bool(nr,1,jj)+
     *           suml(2)*bool(nr,2,jj)+suml(3)*bool(nr,3,jj))
170          continue
             endif
4060        continue

            do 4080 k=1,nrodsuc(j)
             nr=nrodunc(j,k)
             nj=nnodbet(nrodunc(j,k))+2
             if(conmat(nrodunc(j,k),1).eq.nnodunc(j)) then
               rr(nr,1,1)=sum(1)+fx(nnodunc(j))
               rr(nr,1,2)=sum(2)+fy(nnodunc(j))
               rr(nr,1,3)=sum(3)+fz(nnodunc(j))
             else
               rr(nr,nj,1)=sum(1)+fx(nnodunc(j))
               rr(nr,nj,2)=sum(2)+fy(nnodunc(j))
               rr(nr,nj,3)=sum(3)+fz(nnodunc(j))
             endif
4080        continue
4050   continue

       do j=1,npre
          do l=1,3
       sum(l)=0D0
  enddo
        do k=1,nrodspre(j)
          nr=nrodpre(j,k)
          nj=nnodbet(nrodpre(j,k))+2
          if(conmat(nrodpre(j,k),1).eq.nnodpre(j)) then
          do jj=1,3
            suml(jj)=f(nr,1,1)*t(nr,1,1,jj)+f(nr,1,2)*t(nr,1,2,jj)+
*              f(nr,1,3)*t(nr,1,3,jj)
      enddo
          do jj=1,3
           sum(jj)=sum(jj)+(suml(1)*bool(nr,1,jj)+
*               suml(2)*bool(nr,2,jj)+suml(3)*bool(nr,3,jj))
      enddo
          else
          do jj=1,3
           suml(jj)=f(nr,nj,1)*t(nr,nj,1,jj)
*             +f(nr,nj,2)*t(nr,nj,2,jj)+f(nr,nj,3)*t(nr,nj,3,jj)
      enddo
          do jj=1,3
           sum(jj)=sum(jj)-(suml(1)*bool(nr,1,jj)+
*              suml(2)*bool(nr,2,jj)+suml(3)*bool(nr,3,jj))
      enddo
          endif
   enddo
       fx(nnodpre(j))=-sum(1)
       fy(nnodpre(j))=-sum(2)
       fz(nnodpre(j))=-sum(3)
 enddo


 Return
 End


      Subroutine rprime(x,bool,rg,r,nnod,delta,rp,nrod,nnodbet)

         Double precision rg(171,99,3),r(171,99,3),delta(171)
        Double precision rp(171,99,3)
        integer nnodbet(171),nrod
        Double precision x(171,99,3),bool(171,3,3)

     do 3010 j=1,nrod
       nj=nnodbet(j)+2
       do 3020 i=1,nj
         r(j,i,1)=(rg(j,i,1)-x(j,1,1))*bool(j,1,1)+
     *           (rg(j,i,2)-x(j,1,2))*bool(j,1,2)+
     *           (rg(j,i,3)-x(j,1,3))*bool(j,1,3)
             r(j,i,2)=(rg(j,i,1)-x(j,1,1))*bool(j,2,1)+
     *           (rg(j,i,2)-x(j,1,2))*bool(j,2,2)+
     *           (rg(j,i,3)-x(j,1,3))*bool(j,2,3)
             r(j,i,3)=(rg(j,i,1)-x(j,1,1))*bool(j,3,1)+
     *           (rg(j,i,2)-x(j,1,2))*bool(j,3,2)+
     *           (rg(j,i,3)-x(j,1,3))*bool(j,3,3)
 3020      Continue
 3010   continue

        do 3005 j=1,nrod
         nj=nnodbet(j)+2
          rp(j,1,1)=(-r(j,3,1)+4D0*r(j,2,1)
     *        -3D0*r(j,1,1))/(2D0*delta(j))
          rp(j,1,2)=(-r(j,3,2)+4D0*r(j,2,2)
     *        -3D0*r(j,1,2))/(2D0*delta(j))
          rp(j,1,3)=(-r(j,3,3)+4D0*r(j,2,3)
     *        -3D0*r(j,1,3))/(2D0*delta(j))
          rp(j,nj,1)=(3D0*r(j,nj,1)-4D0*r(j,nj-1,1)
     *          +r(j,nj-2,1))/(2D0*delta(j))
          rp(j,nj,2)=(3D0*r(j,nj,2)-4D0*r(j,nj-1,2)
     *          +r(j,nj-2,2))/(2D0*delta(j))
          rp(j,nj,3)=(3D0*r(j,nj,3)-4D0*r(j,nj-1,3)
     *          +r(j,nj-2,3))/(2D0*delta(j))
          Do 3000 i=2 , nj-1
            rp(j,i,1)=(r(j,i+1,1)-r(j,i-1,1))/(2D0*delta(j))
            rp(j,i,2)=(r(j,i+1,2)-r(j,i-1,2))/(2D0*delta(j))
            rp(j,i,3)=(r(j,i+1,3)-r(j,i-1,3))/(2D0*delta(j))
 3000     Continue
 3005   continue

        Return
        End


        Subroutine lanteta(rp,lan,nnod
     *          ,teta,nrod,nnodbet,conmat
     *           ,ncons,ntypeco,nrodsco,nrodcon,nnodcon
     *           ,nuncons,ntypeun,nrodsuc,nrodunc,nnodunc
     *          ,delta,teta0,itype
     + ,nrodspre,ntypepre,nnodpre,nrodpre,npre)


          Double precision rp(171,99,3),lan(171,99)
          Double precision teta(171,99,3)
        Double precision delta(171),teta0
        integer nnodbet(171),nrod
    integer ncons,nuncons,conmat(171,2)
    integer nrodsco(45),ntypeco(45),nnodcon(45)
    integer nrodcon(45,9)
    integer nrodsuc(55),ntypeun(55),nnodunc(55)
    integer nrodunc(55,11)
   integer nrodspre(55),ntypepre(55),nnodpre(55)
   integer nrodpre(55,11),npre


    do 4005 j=1,nrod
        nj=nnodbet(j)+2
         Do 4000 i=2 , nj-1
           lan(j,i)=Sqrt(rp(j,i,1)**2D0+rp(j,i,2)**2D0
     *              +rp(j,i,3)**2D0)
          teta(j,i,2)=Asin(-rp(j,i,3)/lan(j,i))
          teta(j,i,3)=Atan(rp(j,i,2)/rp(j,i,1))
4000      Continue
          lan(j,1)=Sqrt(rp(j,1,1)**2D0+rp(j,1,2)**2D0
     *             +rp(j,1,3)**2D0)
          lan(j,nj)=Sqrt(rp(j,nj,1)**2D0+rp(j,nj,2)**2D0
     *             +rp(j,nj,3)**2D0)
4005  continue

    do 4020 j=1,ncons
      if (ntypeco(j).eq.1) then

         do 4030 k=1,nrodsco(j)
          if(conmat(nrodcon(j,k),1).eq.nnodcon(j)) then
            teta(nrodcon(j,k),1,1)=0D0
            teta(nrodcon(j,k),1,2)=0D0
            teta(nrodcon(j,k),1,3)=0D0
          else
            teta(nrodcon(j,k),nnodbet(nrodcon(j,k))+2,1)=0D0
            teta(nrodcon(j,k),nnodbet(nrodcon(j,k))+2,2)=0D0
            teta(nrodcon(j,k),nnodbet(nrodcon(j,k))+2,3)=0D0
          endif
4030     continue
      else

         do 4040 k=1,nrodsco(j)
           nr=nrodcon(j,k)
           nj=nnodbet(nrodcon(j,k))+2
          if(conmat(nrodcon(j,k),1).eq.nnodcon(j)) then
            teta(nr,1,2)=(-teta(nr,3,2)+4D0*teta(nr,2,2))/3D0
            teta(nr,1,3)=(-teta(nr,3,3)+4D0*teta(nr,2,3))/3D0
           else
            teta(nr,nj,2)=(-teta(nr,nj-2,2)
     *              +4D0*teta(nr,nj-1,2))/3D0
            teta(nr,nj,3)=(-teta(nr,nj-2,3)
     *              +4D0*teta(nr,nj-1,3))/3D0
           endif
4040     continue
        endif
4020  continue

      do 4050 j=1,nuncons
        if (ntypeun(j).eq.2) then
         do 4070 k=1,nrodsuc(j)
            nr=nrodunc(j,k)
            nj=nnodbet(nrodunc(j,k))+2
           if(conmat(nrodunc(j,k),1).eq.nnodunc(j)) then
            teta(nr,1,2)=(-teta(nr,3,2)+4D0*teta(nr,2,2))/3D0
            teta(nr,1,3)=(-teta(nr,3,3)+4D0*teta(nr,2,3))/3D0
           else
            teta(nr,nj,2)=(-teta(nr,nj-2,2)
     *              +4D0*teta(nr,nj-1,2))/3D0
            teta(nr,nj,3)=(-teta(nr,nj-2,3)
     *              +4D0*teta(nr,nj-1,3))/3D0
           endif
4070     continue
        endif
4050  continue


      do j=1,npre
        if (ntypepre(j).eq.2) then
         do k=1,nrodspre(j)
            nr=nrodpre(j,k)
            nj=nnodbet(nrodpre(j,k))+2
           if(conmat(nrodpre(j,k),1).eq.nnodpre(j)) then
            teta(nr,1,2)=(-teta(nr,3,2)+4D0*teta(nr,2,2))/3D0
            teta(nr,1,3)=(-teta(nr,3,3)+4D0*teta(nr,2,3))/3D0
           else
            teta(nr,nj,2)=(-teta(nr,nj-2,2)
     *              +4D0*teta(nr,nj-1,2))/3D0
            teta(nr,nj,3)=(-teta(nr,nj-2,3)
     *              +4D0*teta(nr,nj-1,3))/3D0
           endif
       enddo
        endif
      enddo
      Return
      End


   subroutine trans(rp,lan,nrod,nnodbet,teta,t)

       Double precision teta(171,99,3)
      Double precision t(171,99,3,3)
       Double precision rp(171,99,3),lan(171,99)

       integer nnodbet(171),nrod

   do 4090 j=1,nrod
      nj=nnodbet(j)+2
      Do 4100 i=1 , nj
       t(j,i,1,1)=Cos(teta(j,i,2))*Cos(teta(j,i,3))
       t(j,i,1,2)=Cos(teta(j,i,2))*Sin(teta(j,i,3))
       t(j,i,1,3)=-Sin(teta(j,i,2))
       t(j,i,2,1)=Cos(teta(j,i,3))*Sin(teta(j,i,1))
     *         *Sin(teta(j,i,2))-
     *         Cos(teta(j,i,1))*Sin(teta(j,i,3))
       t(j,i,2,2)=Cos(teta(j,i,1))*Cos(teta(j,i,3))+
     *         Sin(teta(j,i,1))*Sin(teta(j,i,2))
     *         *Sin(teta(j,i,3))
       t(j,i,2,3)=Cos(teta(j,i,2))*Sin(teta(j,i,1))
       t(j,i,3,1)=Cos(teta(j,i,1))*Cos(teta(j,i,3))
     *         *Sin(teta(j,i,2))+
     *         Sin(teta(j,i,1))*Sin(teta(j,i,3))
       t(j,i,3,2)=-Cos(teta(j,i,3))*Sin(teta(j,i,1))+
     *         Cos(teta(j,i,1))*Sin(teta(j,i,2))
     *         *Sin(teta(j,i,3))
       t(j,i,3,3)=Cos(teta(j,i,1))*Cos(teta(j,i,2))
4100   Continue
4090  continue

      Return
      End


      Subroutine tetaprime(lan,r,nnod,delta,tetap,tetapp
     *               ,teta,nrod,nnodbet)
        Double precision lan(171,99),r(171,99,3)
        Double precision tetap(171,99,3),delta(171)
        Double precision teta(171,99,3),tetapp(171,99,3)
        integer nnodbet(171),nrod

      do 5005 j=1,nrod
```

```
        nj=nnodbet(j)+2
         Do 5000 i=1 , nj
           If (i.eq.1) Then
            do 50 k=1,3
                tetap(j,i,k)=(-teta(j,3,k)+4d0*teta(j,2,k)
     *            -3d0*teta(j,1,k))/(2d0*delta(j))
50         continue
           Else if (i.eq.nj) Then
            do 60 k=1,3
                tetap(j,i,k)=(3d0*teta(j,i,k)-4d0*teta(j,i-1,k)
     *              +teta(j,i-2,k))/(2d0*delta(j))
60         continue
           Else
            do 70 k=1,3
                tetap(j,i,k)=(teta(j,i+1,k)-teta(j,i-1,k))
     *              /(2d0*delta(j))
70         continue
           End If
5000     Continue
5005   continue

       do 6005 j=1,nrod
        nj=nnodbet(j)+2
        Do 6000 i=1 , nj
         do 80 k=1,3
           If (i.eq.1) Then
              tetapp(j,i,k)=(-teta(j,4,k)+4d0*teta(j,3,k)
     *          -5d0*teta(j,2,k)
     *          +2d0*teta(j,1,k)) /(delta(j)**2d0)
           Elseif (i.eq.nj) then
            tetapp(j,i,k)=(-teta(j,nj-3,k)+4d0*teta(j,nj-2,k)
     *         -5d0*teta(j,nj-1,k)
     *         +2d0*teta(j,nj,k))/(delta(j)**2d0)
          else
              tetapp(j,i,k)=(teta(j,i+1,k)-2d0*teta(j,i,k)
     *          +teta(j,i-1,k))/(delta(j)**2d0)
           End If
80      continue
6000    Continue
6005   continue

       Return
       End


    subroutine kapaprime(kapa,kapap,nrod,nnodbet
     *          ,teta,tetap,tetapp)
```

```
      Double precision teta(171,99,3),tetap(171,99,3)
      Double precision tetapp(171,99,3)
      Double precision kapa(171,99,3),kapap(171,99,3)

      integer nnodbet(171),nrod

      do 4090 j=1,nrod
         nj=nnodbet(j)+2
         Do 4100 i=1 , nj
           kapa(j,i,1)=tetap(j,i,1)-tetap(j,i,3)
     *         *Sin(teta(j,i,2))
           kapa(j,i,2)=tetap(j,i,2)*Cos(teta(j,i,1))+
     *           tetap(j,i,3)*Sin(teta(j,i,1))
     *         *Cos(teta(j,i,2))
           kapa(j,i,3)=-tetap(j,i,2)*Sin(teta(j,i,1))+
     *           tetap(j,i,3)*Cos(teta(j,i,1))
     *         *Cos(teta(j,i,2))
           kapap(j,i,1)=tetapp(j,i,1)-tetapp(j,i,3)
     *         *Sin(teta(j,i,2))-tetap(j,i,3)
     *         *tetap(j,i,2)*Cos(teta(j,i,2))
           kapap(j,i,2)=tetapp(j,i,2)*Cos(teta(j,i,1))
     *         -tetap(j,i,2)*tetap(j,i,1)*Sin(teta(j,i,1))
     *         +tetapp(j,i,3)*Sin(teta(j,i,1))*Cos(teta(j,i,2))
     *         +tetap(j,i,3)*tetap(j,i,1)*Cos(teta(j,i,1))
     *         *Cos(teta(j,i,2))
     *         -tetap(j,i,3)*tetap(j,i,2)*Sin(teta(j,i,1))
     *         *Sin(teta(j,i,2))
           kapap(j,i,3)=-tetapp(j,i,2)*Sin(teta(j,i,1))
     *         -tetap(j,i,2)*tetap(j,i,1)*Cos(teta(j,i,1))
     *         +tetapp(j,i,3)*Cos(teta(j,i,1))*Cos(teta(j,i,2))
     *         -tetap(j,i,3)*tetap(j,i,1)*Sin(teta(j,i,1))
     *         *Cos(teta(j,i,2))
     *         -tetap(j,i,3)*tetap(j,i,2)*Cos(teta(j,i,1))
     *         *Sin(teta(j,i,2))

4100   Continue
4090  continue


        Return
        End


        Subroutine calcf(lan,nnod,e,kapa,kapap
     *           ,jx,gs,area,nrod,nnodbet
     *           ,iz,delta,f,sigmay,x1,x2,length
     *          ,itypeprob
     *        ,epsilonold,sigmaold,icodeold
     *        ,epsilonnew,sigmanew,icodenew,iconverged,e2,e3)


      Double precision epsilonnew(171,99),sigmanew(171,99)
      integer icodenew(171,99)
      Double precision e2(171),e3(171)
      Double precision epsilonold(171,99),sigmaold(171,99)
      Double precision sigmay(171),x1,x2,length(171)
          Double precision lan(171,99)
          Double precision e(171),area(171)
          Double precision gs(171),jx(171)
          Double precision iz(171),delta(171)
         Double precision kapa(171,99,3),kapap(171,99,3)
         Double precision mprime(171,99,3),f(171,99,3)
         integer icodeold(171,99)
          integer nnodbet(171),nrod

       do 1005 j=1,nrod
          nj=nnodbet(j)+2
           Do 1000 i=1 , nj
            mprime(j,i,1)=gs(j)*jx(j)*kapap(j,i,1)

            mprime(j,i,2)=e(j)*iz(j)*kapap(j,i,2)
     *              -e(j)*iz(j)*kapa(j,i,1)*kapa(j,i,3)
     *              +gs(j)*jx(j)*kapa(j,i,1)*kapa(j,i,3)

            mprime(j,i,3)=e(j)*iz(j)*kapap(j,i,3)
     *              +e(j)*iz(j)*kapa(j,i,1)*kapa(j,i,2)
     *              -gs(j)*jx(j)*kapa(j,i,1)*kapa(j,i,2)
1000        Continue
1005   continue

       do 4005 j=1,nrod
          nj=nnodbet(j)+2
           Do 4000 i=1 , nj
            f(j,i,2)=-mprime(j,i,3)/lan(j,i)

            f(j,i,3)=mprime(j,i,2)/lan(j,i)
4000        Continue
4005   continue


      if(itypeprob.eq.1)then
       do j=1,nrod
        nj=nnodbet(j)+2
         Do i=1 , nj
       f(j,i,1)=e(j)*area(j)*(lan(j,i)-1D0)
       epsilonnew(j,i)=(lan(j,i)-1D0)
       sigmanew(j,i)=e(j)*(lan(j,i)-1D0)
      Enddo
   enddo

  elseif(itypeprob.eq.2)then

    Call constitutive2(lan,e,area,length
*            ,iz,nrod,nnodbet,f,x1,x2,sigmay,iprint
*           ,epsilonold,sigmaold,icodeold
*           ,epsilonnew,sigmanew,icodenew)

  elseif(itypeprob.eq.3)then

    Call constitutive3(lan,e,area,length
*            ,iz,nrod,nnodbet,f,x1,x2,sigmay,iprint
*           ,epsilonold,sigmaold,icodeold
*           ,epsilonnew,sigmanew,icodenew)

  elseif(itypeprob.eq.4)then

    Call constitutive4(lan,e,area,length
*            ,iz,nrod,nnodbet,f,x1,x2,sigmay,0
*           ,epsilonold,sigmaold,icodeold
*           ,epsilonnew,sigmanew,icodenew,iconverged,e2,e3)

  endif

        Return
        End

    Subroutine fprime(f,fp,delta,nrod,nnodbet)

       Double precision f(171,99,3),fp(171,99,3),delta(171)
        integer nnodbet(171),nrod


  do 6015 j=1,nrod
   nj=nnodbet(j)+2
   Do 6010 i=1 , nj
    do 10 k=1,3
      if(i.eq.1) then
        fp(j,1,k)=(-f(j,3,k)+4d0*f(j,2,k)-3d0*f(j,1,k))
*            /(2d0*delta(j))
      elseif(i.eq.nj) then
        fp(j,nj,k)=(3d0*f(j,nj,k)-4d0*f(j,nj-1,k)
*             +f(j,nj-2,k))/(2d0*delta(j))
```

```
        else
           fp(j,i,k)=(f(j,i+1,k)-f(j,i-1,k))/(2d0*delta(j))
        endif
10    continue
6010  Continue
6015 continue

        Return
        End


     subroutine transprime(nrod,nnodbet,teta,tetap,tp)

        Double precision teta(171,99,3),tetap(171,99,3)
      Double precision tp(171,99,3,3)

        integer nnodbet(171),nrod


     do 4090 j=1,nrod
        nj=nnodbet(j)+2
        Do 4100 i=1 , nj

          tp(j,i,1,1)=-tetap(j,i,2)*Cos(teta(j,i,3))*
     *         Sin(teta(j,i,2))-
     *          tetap(j,i,3)*Cos(teta(j,i,2))*
     *         Sin(teta(j,i,3))

          tp(j,i,1,2)=-tetap(j,i,2)*Sin(teta(j,i,2))*
     *         Sin(teta(j,i,3))+
     *          tetap(j,i,3)*Cos(teta(j,i,2))*
     *         Cos(teta(j,i,3))

          tp(j,i,1,3)=-tetap(j,i,2)*Cos(teta(j,i,2))


          tp(j,i,2,1)=tetap(j,i,1)*Cos(teta(j,i,1))*
     *         Cos(teta(j,i,3))*Sin(teta(j,i,2))+
     *          tetap(j,i,1)*Sin(teta(j,i,1))*
     *         Sin(teta(j,i,3))+
     *          tetap(j,i,2)*Cos(teta(j,i,2))*
     *         Cos(teta(j,i,3))*Sin(teta(j,i,1))-
     *          tetap(j,i,3)*Cos(teta(j,i,1))*
     *         Cos(teta(j,i,3))-
     *          tetap(j,i,3)*Sin(teta(j,i,2))*
     *         Sin(teta(j,i,3))*Sin(teta(j,i,1))
```

```
        tp(j,i,2,2)=-tetap(j,i,1)*Cos(teta(j,i,3))*
     *          Sin(teta(j,i,1))+
     *           tetap(j,i,1)*Cos(teta(j,i,1))*
     *          Sin(teta(j,i,2))*Sin(teta(j,i,3))+
     *           tetap(j,i,2)*Cos(teta(j,i,2))*
     *          Sin(teta(j,i,1))*Sin(teta(j,i,3))+
     *           tetap(j,i,3)*Cos(teta(j,i,3))*
     *          Sin(teta(j,i,1))*Sin(teta(j,i,2))-
     *           tetap(j,i,3)*Cos(teta(j,i,1))*
     *          Sin(teta(j,i,3))

        tp(j,i,2,3)=tetap(j,i,1)*Cos(teta(j,i,1))*
     *          Cos(teta(j,i,2))-
     *           tetap(j,i,2)*Sin(teta(j,i,1))*
     *          Sin(teta(j,i,2))


        tp(j,i,3,1)=-tetap(j,i,1)*Cos(teta(j,i,3))*
     *          Sin(teta(j,i,1))*Sin(teta(j,i,2))+
     *           tetap(j,i,1)*Cos(teta(j,i,1))*
     *          Sin(teta(j,i,3))+
     *           tetap(j,i,2)*Cos(teta(j,i,1))*
     *          Cos(teta(j,i,2))*Cos(teta(j,i,3))+
     *           tetap(j,i,3)*Cos(teta(j,i,3))*
     *          Sin(teta(j,i,1))-
     *           tetap(j,i,3)*Cos(teta(j,i,1))*
     *          Sin(teta(j,i,2))*Sin(teta(j,i,3))

        tp(j,i,3,2)=-tetap(j,i,1)*Cos(teta(j,i,1))*
     *          Cos(teta(j,i,3))-
     *           tetap(j,i,1)*Sin(teta(j,i,1))*
     *          Sin(teta(j,i,2))*Sin(teta(j,i,3))+
     *           tetap(j,i,2)*Cos(teta(j,i,1))*
     *          Cos(teta(j,i,2))*Sin(teta(j,i,3))+
     *           tetap(j,i,3)*Cos(teta(j,i,1))*
     *          Cos(teta(j,i,3))*Sin(teta(j,i,2))+
     *           tetap(j,i,3)*Sin(teta(j,i,1))*
     *          Sin(teta(j,i,3))

        tp(j,i,3,3)=-tetap(j,i,1)*Cos(teta(j,i,2))*
     *          Sin(teta(j,i,1))-
     *           tetap(j,i,2)*Cos(teta(j,i,1))*
     *          Sin(teta(j,i,2))


4100    Continue
4090  continue
```

```
      Return
      End




      Subroutine output(lan,no,r,nnod,nrod,nnodbet
     * ,cu,mu,cv,mv,cw,mw,error,deltat,x,jf)

   double precision cu,mu
   double precision cv,mv,cw,mw
   double precision error,deltat

        Double precision r(171,99,3),lan(171,99),x(171,99,3)
       integer nnodbet(171),nrod

      if(jf.eq.1)then
       nnode=0
        nconmat=0
       do 1000 j=1,nrod
         nnode=nnode+nnodbet(j)+2
         nconmat=nconmat+nnodbet(j)+1
1000    continue
       write(2,*) nnode
       do 1100 j=1,nrod
         nj=nnodbet(j)+2
         do 1200 i=1,nj
           write(2,*) x(j,i,1),x(j,i,2),x(j,i,3)
1200      continue
1100    continue

       write(2,*) nconmat
       do 1300 j=1,nrod
         nj=nnodbet(j)+2
         do 1400 i=1,nj-1
          n1=i+(j-1)*(nnodbet(j-1)+2)
          n2=i+1+(j-1)*(nnodbet(j-1)+2)
           write(2,*) n1,n2
1400      continue
1300    continue

      endif
       do 1500 j=1,nrod
         nj=nnodbet(j)+2
         do 1600 i=1,nj
           write(2,*) r(j,i,1),r(j,i,2),r(j,i,3)
```

```
1600       continue
1500     continue

         if(jf.eq.1)then
          write(2,*)'No. of iterations=',no
           write(2,*)'cu,mu=',cu,mu
           write(2,*)'cv,mv=',cv,mv
            write(2,*)'cw,mw=',cw,mw
          write(2,*)'error,deltat=',error,deltat
      endif
     do 7005 j=1,nrod
         nj=nnodbet(j)+2
          Do 7000 i=1 , nj
c             write(2,*)j,i,r(j,i,1),r(j,i,2),r(j,i,3)
7000       Continue
7005   continue

     do 7110 j=1,nrod
         nj=nnodbet(j)+2
          Do 7100 i=1 , nj
c            write(2,*)'lambda',j,i,lan(j,i)
7100       Continue
7110   continue

        Return
        End
```